USACERL Technical Report FE-94/12
June 1994

Development and Field Test of the Central Energy Plant Adaptive Control System (CEPACS)

by
Mike C.J. Lin
Yaoxin Qian
Dow Drake
Norman Miller
Joseph Bentsman

Spare parts for the outdated control systems at many Army central energy plants (CEPs) are difficult to acquire. This fact, combined with recent price reductions in modern control hardware and software, has made it practical to replace these old systems with controls that incorporate modern electronics and advanced boiler-control technologies to improve safety, reliability, and efficiency.

This report documents the development of a prototype Central Energy Plant Adaptive Control System (CEPACS) and the results of a field test of the system in a gas/oil-fired boiler at the University of Illinois Abbott Power Plant in Champaign, IL. The prototype self-tuning adaptive system is a complete controller for a steam generation system that achieves a high degree of operational automation; the system significantly reduces the need for operator intervention into plant operation. The first set of field trials demonstrated excellent combustion control as well as improved drum pressure controls. The results indicate immediate payback in terms of fuel savings and enhanced safety.

Approved for public release; distribution is unlimited.

USER EVALUATION OF REPORT

REFERENCE: USACERL Technical Report FE-94/12, *Development and Field Test of the Central Energy Plant Adaptive Control System (CEPACS)*

Please take a few minutes to answer the questions below, tear out this sheet, and return it to USACERL. As user of this report, your customer comments will provide USACERL with information essential for improving future reports.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

2. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

____

____

3. Has the information in this report led to any quantitative savings as far as manhours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

4. What is your evaluation of this report in the following areas?

a. Presentation: ______________________________

b. Completeness: ______________________________

c. Easy to Understand: ______________________________

d. Easy to Implement: ______________________________

e. Adequate Reference Material: ______________________________

f. Relates to Area of Interest: ______________________________

g. Did the report meet your expectations? ______________________________

h. Does the report raise unanswered questions? ______________________________

i. General Comments. (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

____

____

____

____

____

5. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ____________________

Telephone Number: ____________________

Organization Address: ____________________

____

____

6. Please mail the completed form to:

Department of the Army
CONSTRUCTION ENGINEERING RESEARCH LABORATORIES
ATTN: CECER-IMT
P.O. Box 9005
Champaign, IL 61826-9005

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1994	Final

4. TITLE AND SUBTITLE
Development and Field Test of the Central Energy Plant Adaptive Control System (CEPACS)

5. FUNDING NUMBERS
4A162784
AT45
EAX22

6. AUTHOR(S)
Mike C.J. Lin, Yaoxin Qian, Dow Drake, Norman Miller, and Joseph Bentsman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Construction Engineering Research Laboratories (USACERL)
P.O. Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
TR FE-94/12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Center for Public Works (USACPW)
ATTN: CECPW-FU-M
7701 Telegraph Road
Alexandria, VA 22310-3862

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Spare parts for the outdated control systems at many Army central energy plants (CEPs) are difficult to acquire. This fact, combined with recent price reductions in modern control hardware and software, has made it practical to replace these old systems with controls that incorporate modern electronics and advanced boiler-control technologies to improve safety, reliability, and efficiency.

This report documents the development of a prototype Central Energy Plant Adaptive Control System (CEPACS) and the results of a field test of the system in a gas/oil-fired boiler at the University of Illinois Abbott Power Plant in Champaign, IL. The prototype self-tuning adaptive system is a complete controller for a steam generation system that achieves a high degree of operational automation; the system significantly reduces the need for operator intervention into plant operation. The first set of field trials demonstrated excellent combustion control as well as improved drum pressure controls. The results indicate immediate payback in terms of fuel savings and enhanced safety.

14. SUBJECT TERMS
Central Energy Plant Adaptive Control System (CEPACS)
gas-fired heating plants
oil-fired heating plants
cost effectiveness
energy efficient

15. NUMBER OF PAGES
118

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

FOREWORD

This study was conducted for the U.S. Army Center for Public Works (USACPW), under project 4A1627814AT45, "Energy and Energy Conservation"; Work Unit EA-X22, "Central Plant Controls." The technical monitor was Philip Conner, CECPW-FU.

The research was performed by the Fuel and Power Systems Team (FEP), Energy and Utility Systems Division (FE), Infrastructure Laboratory (FL), U.S. Army Construction Engineering Research Laboratories (USACERL). Dr. Mike C.J. Lin was the USACERL principal investigator. Special acknowledgement is given to the following Abbott Power Plant personnel for their support and cooperation in the testing of the prototype system: Daniel Dancey, Donald Fortik, Donald White, and other instrument technicians and boiler operators too numerous to mention. Yaoxin Qian, Dow Drake, Dr. Norman Miller, and Dr. Joseph Bentsman are with the Department of Industrial and Mechanical Engineering, University of Illinois, Urbana-Champaign. Gary Schanche is Team Leader, CECER-FEP. Dr. David Joncich is Chief, CECER-FE, and Dr. Michael J. O'Connor is Chief, CECER-FL. The USACERL technical editor was William J. Wolfe, Information Management Office.

LTC David J. Rehbein is Commander and Acting Director, USACERL. Dr. Michael J. O'Connor is Technical Director.

CONTENTS

CONTENTS (Cont'd)

FIGURES

FIGURES (Cont'd)

FIGURES (Cont'd)

TABLES

DEVELOPMENT AND FIELD TEST OF THE CENTRAL ENERGY PLANT ADAPTIVE CONTROL SYSTEM (CEPACS)

1 INTRODUCTION

Background

U.S. Army installations operate and maintain a large number of fossil fuel-powered central energy plants (CEPs). A general problem at Army CEPs is that outdated plant control systems have become both economically and operationally inefficient when compared to modern systems. A reliable automatic control system is an essential element in maintaining safe, efficient CEP operation. The difficulty of acquiring spare parts for these old systems, and the recent price reduction in modern control hardware and software have made it more practical to replace the obsolete or primitive control systems used in many Army facilities with controls that incorporate state-of-the-art electronics and advanced boiler-control technologies to improve safety, reliability, and efficiency.

The U.S. Army Construction Engineering Research Laboratories (USACERL) and the University of Illinois Mechanical Engineering Department are developing and testing a prototype advanced, self-tuning Central Energy Plant Adaptive Control System (CEPACS) for retrofit to old systems or installation in new boiler applications. This prototype system promises to reduce the need for skilled operators since it captures plant operating experience and retunes itself for optimal performance. This application uses a control strategy based on the Generalized Predictive Control (GPC) algorithm (Clarke, Mohtadi, and Tuffs 1987), which has been widely applied in a number of process industries in the United Kingdom.

Traditional boiler control methods base their control schemes on the general assumption that plant dynamics are static through time. The GPC algorithm assumes that plant dynamics are in flux; it updates the plant model parameters after each sampling period, continuously modifying the control strategy as needed to bring the plant to the desired set of conditions in the minimum amount of time. The model automatically adjusts itself to changes in equipment configuration or behavior as well as to ambient conditions. Applying the GPC to boiler control is a new concept that needs to be field-tested as a first step toward implementing of the CEPACS system. This field test of CEPACS was designed to determine the effectiveness of the self-tuning control system in a CEP environment.

Objectives

The objectives of this project were to develop and field test a prototype Central Energy Plant Adaptive Control System to be installed in Army facilities to improve plant safety, reliability, and efficiency.

Approach

Literature on boiler control basics was reviewed and analyzed. The prototype self-tuning adaptive control system was developed, including the control algorithm and the required hardware and software. A field test was conducted on a gas- and oil-fired boiler at the University of Illinois Abbott Power Plant in July 1991. Test results were analyzed, and further system enhancements such as robustification, faults tolerance, and diagnostics were explored. Such essential processes as the automatic startup/shutdown procedures, furnace draft control, and on-line calculation of the boiler combustion efficiency were implemented and documented.

Scope

CEPACS is generally applicable to gas- or oil-fired boilers. Solid fuel fired units, however, require modifications of the control strategies, mainly due to the difficulties in accurately measuring and controlling the flow rate of the solid fuel.

Mode of Technology Transfer

It is recommended that the information presented in this report be incorporated into Design Guide (DG) and an Engineering Technical Note (ETN).

2 BOILER CONTROL BASICS AND BOILER MODELS

Depending on system complexity, CEPs use any of several different types of control: (1) on/off control (also called two-position control); (2) proportional control (also called gain control or throttling control); (3) integral control (also called reset control); (4) derivative control (also called rate control); and (5) proportional-integral-derivative control (also called PID, or three-mode control).

The use of conventional proportional, integral, and derivative controllers may not result in the highest possible operating efficiency because of the large variations in plant inputs and disturbances affecting the process. In boilers, these variations typically arise from changes in fuel calorific value, warmup effects in the burner, fouling of the boiler tubes, or deterioration of mechanical links and profiles involved in the operation. Climatic changes in ambient temperature, pressure, and humidity can also affect the control adversely. For the control system to deal with such changes effectively, the controller must be tuned in real time, preferably without operator intervention. This can be done by implementing a loop between the control signal and the output of the process to create a relationship between the two. This implies the generation of a process model with parameters that are estimated recursively. These parameters are then used in a design control algorithm to compute optimum gains for the controller. The ability to estimate model parameters in real time for a range of loads and inputs results in online tuning of the controller gains.

Since self-tuning is faster than manual tuning, the commissioning time for installation can be decreased by as much as 50 percent. Also, self-tuning performs its processes systematically, even for the simplest control loops. Difficulties have been encountered, however, with processes with very rapid parameter variations, or very strong nonlinearities. The controller cannot be applied to processes that do not tolerate the process setpoint deviation required in the identification phase. Difficulties have also been found under operating conditions where the measured value is suddenly disconnected. The remedy in such cases is to stop the parameter updating and switch the controller to operate in the conventional PID mode.

PID controllers implemented in pneumatic, electronic, or microcomputer software can be somewhat difficult to set up and may not produce stable results over a wide range of operating conditions. Adaptive controllers are aimed at overcoming these deficiencies, which originated from dealing with control problems related to high performance aircraft and rockets. A review of recent adaptive control algorithms indicated that the GPC algorithm was the algorithm most able to meet the control requirements of CEPS. Researchers developed a multivariable mathematical model for an industrial boiler and a real time simulator for a steam generation system to evaluate various control algorithms and operator training. This model was tailored to fit an actual boiler, and the tailored model was then used to redesign the original GPC-based control system. The system also contained modifications to the safety interlocks and feed-forward control, and used the identifier information extensively.

In enhancing the mathematical model of the boiler, researchers examined many analytical models that had been derived to predict the behavior of heating plant boilers. Some of the relevant models recently developed were compared and analyzed to determine their suitability for application (Åström and Bell 1988; Bell and Åström 1987; McDonald, Kwatney, and Spare 1971; and Chawdry and Hogg 1989).

Åström and Bell (1987) compared several previously developed models (Morton and Price 1977; Åström and Eklund 1972, 1975; Bell and Åström 1987), ranging from simple linear to nonlinear models of varying degrees. The study analyzed the most effective part of each previous model and combined them into a single nonlinear model that used a combination of first principles and field data to determine model parameters. The plant studied was a 160 MW oil-fired natural circulation drum unit operated by

Sydkraft AB in Malmo, Sweden. The plant was rated at 1,108,800 lb/hr steam at 1989 psig.* The inputs to the model were fuel flow, feedwater flow, and control valve position. The plant outputs were drum pressure, drum water level, steam flow rate, and electrical output.

This nonlinear model structure is very similar to the required form of the research system. The electrical output subsystem was uncoupled from the rest of the equations, so separating it from the remaining part of the system was simple. The oxygen level subsystem could be added easily to the equations. Steam flow was designated as an output driven by state equations and an imaginary value representing the net sum of the steam use, which could be manipulated into the test boiler configuration.

To suit the research system, several modifications and additions were made to the first Åström and Bell model (1987). An equation governing the oxygen level loop was developed and added to the system. The equation considers a stoichiometric reaction of fuel and air with a first-order lag. Additional regime is also needed during the startup and shutdown periods, where the boiler passes through several different regimes. By altering the constants in the model to cover a wider range of conditions while still meeting the steady state operation model at the operating point, a general model to go from the start to the steady state can be obtained. It is also possible to account for the fact that the drum is circular. For normal operation, the behavior of the water level is close to linear, but if the water level approaches the extremes of the drum, the nonlinearities become significant. A solution to the nonlinear equation is very difficult to find, but by fitting a polynomial to the equation, a relationship between the change in height and the change in volume can be derived. One final change in the model was to add actuator dynamics to the input signals. These dynamics can be modeled by a pure time delay and a computation lag. The boiler simulation models in state space structure are:

$$\begin{aligned}
\dot{x}_1 &= c_{11} x_1^{\frac{9}{8}} \; c_{12} u_4 x_1^{\frac{1}{8}} + c_{12} u_1 \; c_{13} u_3 \\
\dot{x}_2 &= \frac{c_{21} u_2 \; c_{22} u_1}{u_1 + u_2} \; c_{23} x_2 \\
\dot{x}_3 &= c_{31} x_1 \; c_{32} u_4 \qquad\qquad \text{[Eq 1]} \\
y_1 &= c_{51} x_1 \\
y_2 &= x_2 \\
y_3 &= \frac{(c_{41} x_2 + 1)(x_1 + c_{42})}{x_2 x_1 + c_{43})} + (c_{44} u_1 \; c_{45} u_2 + c_{46})(u_4 c_{47} x_1)
\end{aligned}$$

Where

- x_1 is the pressure
- x_2 is the oxygen level
- x_3 is the fluid density
- u_1 is the fuel flow
- u_2 is the air flow
- u_3 is the feedwater flow
- u_4 is the steam flow.

*1 lb = 0.453 kg; 1 psi = 6.89 kPa.

The outputs y_1, y_2, and y_3 are drum pressure, oxygen level, and water level, respectively. All the c_{ij}'s are the proper scaling constants.

For control purpose, a linear boiler model is needed, and can be reasonably obtained by performing the linearization of the nonlinear model around the boiler operating region. The boiler model in the linear state space form is:

$$\begin{aligned}
\dot{x}_1 &= A_{11}x_1 + B_{11}u_1 + B_{13}u_3 + B_{14}u_4 \\
\dot{x}_2 &= A_{22}x2 + B_{21}u_1 + B_{22}u_2 \\
\dot{x}_3 &= B_{33}u_3 + B_{34}u_4 \\
y_1 &= C_{11}x_1 \\
y_2 &= C_{22}x_2 \\
y_3 &= C_{31}x_1 + C_{33}x_3 + D_{31}u_1 + D_{33}u_3 + D_{34}u_4
\end{aligned} \qquad \text{[Eq 2]}$$

Where A_{ij}'s, B_{ij}'s, C_{ij}'s and D_{ij}'s are the constants determined by the experiments.

The test boiler was a dual-fuel (oil and gas) fired unit capable of producing 175,000 lb/hr of steam at a pressure of 325 psig. The steam from this no. 2 boiler and the other gas boilers flows into a common header with the steam throttled by a turbine and a Republic valve in normal operation. Researchers were interested in controlling the header pressure, the level of the water in the drum, and the oxygen level in the flue gas. These outputs should be maintained despite variations in steam demand, fluctuations in the heating value of the fuel, or other disturbances to the system. To meet these control objectives, the control system presently installed is capable of actuating gas flow, oil flow, air flow, and feedwater flow while sensing steam pressure, steam temperature, steam flow, fuel flow, drum water level, feedwater flow, air flow, and flue gas oxygen level. To improve the boiler model to match the actual plant, operating data was collected from the plant. This required the interface of the developed software with plant equipment, the actual data collection process, and data manipulation. It further required altering the nonlinear equations to be consistent with these results. The validity of the final model was verified through simulations (Appendix A).

Based on physical principles and the analysis of real test data obtained in July 1991 at the Abbott boiler, the recommended model structures for each loop (the model order and corresponding time delay) are:

1. Model for drum pressure loop:

$$A(q^{-1})y(k) = q^{-n}B(q^{-1})u(k) \qquad \text{[Eq 3]}$$

where

$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2}$
$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q\text{-}2$
$y(k)$ is the drum pressure
$u(k)$ is the fuel flow command
k is the sample number
delay n = 10 seconds.

The initial parameters used for this loop are:

$a_1 = -0.8959$
$a_2 = -0.043$
$b_0 = 0.2778$
$b_1 = 0.1503$
$b_2 = 0.01523$

2. Model for excess oxygen loop:

$$A(q^{-1})y(k) = q^{-n_1}B_1(q^{-1})u_1(k) + q^{-n_2}B_2(q^{-1})u_2(k) \qquad \text{[Eq 4]}$$

where
$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3}$
$B_1(q^{-1}) = b_{10} + b_{11}q^{-1} + b_{12}q^{-2}$
$B_2(q^{-1}) = b_{20} + b_{21}q^{-1} + b_{22}q\text{-}2$
$y(k)$ is the excess oxygen level
$u_1(k)$ and $u_2(k)$ are the fuel flow and air flow commands, respectively
k is the sample number
delay n_1=15 seconds
n_2= 10 seconds.

The initial parameter values used are:

$a_1 = -0.562$
$a_2 = 0.207$
$a_3 = -0.0545$
$b_{10} = 0.140$
$b_{11} = 0.0233$
$b_{12} = -0.00723$
$b_{20} = -0.196$
$b_{21} = -0.0659$
$b_{22} = 0.0183$

3. Model for water level loop:

$$A(q^{-1})y(k) = q^{-n_1}B_1(q^{-1})u_1(k) + q^{-n_2}B_2(q^{-1})u_2(k) \qquad \text{[Eq 5]}$$

where
$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3}$
$B_1(q^{-1}) = b_{10} + b_{11}q^{-1} + b_{12}q^{-2} + b_{13}q^{-3}$
$B_2(q^{-1}) = b_{20} + b_{21}q^{-1} + b_{22}q^{-2} + b_{23}q^{-3}$
$y(k)$ is the drum water level
$u_1(k)$ and $u_2(k)$ are the observed steam flow and feed water flow command
k is the sample number and delay n_1 = 20 seconds
n_2 = 20 seconds.

The initial parameter values used are:

$a_1 = -1.3199$
$a_2 = 0.5725$
$a_3 = -0.2518$
$b_{10} = 0.001793$
$b_{11} = 0.001406$
$b_{12} = -0.001742$
$b_{13} = -0.002207$
$b_{20} = -0.00052$
$b_{21} = 0.001479$
$b_{22} = -0.000424$
$b_{23} = 0.000194$

3 PROTOTYPE ADAPTIVE CONTROL SYSTEM

Hardware Description

Microprocessor-based control systems are increasingly used in industrial processes. A wide variety of computerized systems packaged in different forms are available for central heat plant controls. Selecting a system can be a trying process because of the many vendors available and the rapid pace at which digital hardware and software components are changing. In addition, determining the control features and performance objectives requires a thorough analysis based on technical and economical considerations. After much research, a personal computer (PC) based system was chosen based on the system's flexibility and cost (Wohadlo et al. 1990). This system uses a PC with third-party control software combined with front-end input/output (I/O) signal conditioning to perform analog-to-digital and digital-to-analog conversions. System functions include data collection, report generation, PID/advanced control, alarming, and operator interface. The control software selected, *THE FIX*, is produced by Intellution, Inc. The controller used, *μMAC*, was manufactured by Analog Devices, Inc.*

THE FIX software enables the *μMAC* controller to monitor and control industrial processes when used with a DOS-compatible PC. *THE FIX* is a menu-driven, multitasking software package that communicates with distributed I/O devices and provides users with the ability to use a PC workstation as a supervisory host computer. The base configuration includes software for analog and digital I/O, alarm detection and messaging, real-time trending, online calculation blocks and user-drawn color graphics. *THE FIX* is also available in runtime-only systems that are useful for multiple installations.

The μMAC-6000 is a modular I/O processor that offers input and output interfaces, and communications and computing capabilities. This processor can be used as a standalone system, with a host, or as a part of a distributed control system. The base system has 24 analog and 48 digital I/O points, and can be expanded to over 200 analog and 1000 digital I/O points per cluster. The controller central processing unit (CPU) is based on Intel 80188 microprocessor running at a clock rate of 8 MHz. The CPU comes with 256K bytes of CMOS battery-backed RAM memory and 192K bytes of ROM memory. In the C Programmable version, 64K bytes are taken up by the compiler and 128K is left for the user's compiled C program. Application development on the μMAC-6000 is simplified by the use of high-level languages, special libraries of high-level functions, program development aides, and runtime communication support. MCComm+ is the Analog Device's communication protocol, created to provide runtime support from host computers to μMAC systems. Interrupts provide a mechanism for the controller to react to events happening in real time and to help provide structure in programs designed for control of events. Six interrupts are built in the μMAC, all of which are available for program use.

For a prototype control system, issues related to long-term safety and reliability needed to be addressed. Loss of power and software upset could cause the command signals (water flow, air flow, and fuel flow) to be turned off, either singly or as a group. In particular, the software accomplishes fuel/air interlocking. A software crash could cause a loss of air flow while fuel flow continued. A backup system to address these problems had to be implemented before testing. Redundant power supplies were needed to provide DC power to the backup system. A 7312 LOOPMATE manufactured by Control Technology Inc. (5734 Middlebrook Pike, Knoxville, TN, tel 615/584-0440) was chosen as system backup. The LOOPMATE is designed to provide automatic and hard manual backup for process loop control in harsh environments. The LOOPMATE provides a highly visible display of critical sensor and control signal

---

* Intellution, Inc., 315 Norwood Park South, Norwood, MA 02060, tel 617/769-8878; Analog Devices, Inc., One Technology Way, PO Box 9106, Norwood, MA 02060, tel. 617/326-6666. Note that μMac controllers are currently sold by Azonix Corp., 900 Middlesex Turnpike, Building G, Billercia, MA 01821, tel. 1-800/365-1663 (toll-free).

variables and allows optional remote set-point changes and display. During normal operation, the LOOPMATE is transparent to the controller. The LOOPMATE takes over when an out-of-control situation is detected. The user may then select one of the three manual control modes, either to default to the last valid control variable output, ramp to a preset control variable output value, or default to 4 mA or 1 V.

To test the prototype control system in an operating boiler, it was first necessary to obtain detailed existing plant pneumatic control system data, including construction data and schematics, ladder diagrams of electrical safeties, plant control requirements, and control system characteristics. The µMAC controller needs to have current inputs, so pneumatic-to-current converters had to be installed in the control lines. The oxygen level signal was already a 1 to 5 V signal, but the other signals were 3 to 15 psi pneumatic signals. Further processing was necessary on air flow, feedwater flow, and steam flow signals. These signals were measured with differential pressure flowmeters, so a root extractor was necessary to obtain the true flow. Control was switched over from the existing control system to the µMAC by a multiport valve installed by a plant technician. To switch control over from the automatic control to the µMAC, the control system was put in manual mode and the valve was switched. When the µMAC signal and the manual command signal matched to provide a bumpless transfer, the panel was switched back into the automatic mode, allowing the µMAC to take over system control.

GPC Algorithm

Basic Controller Structure

The purpose of any controller is to ensure robust performance. This means that the controller must provide: (1) good tracking of the reference input by plant output, and (2) disturbance rejection, i.e., to maintain tracked values in spite of external disturbances and to preserve such performances under variations in plant characteristics. For a large class of processes, this purpose has been demonstrated to be well served by robust self-tuning controllers. A self-tuning controller (Figure 1) consists of the following blocks:

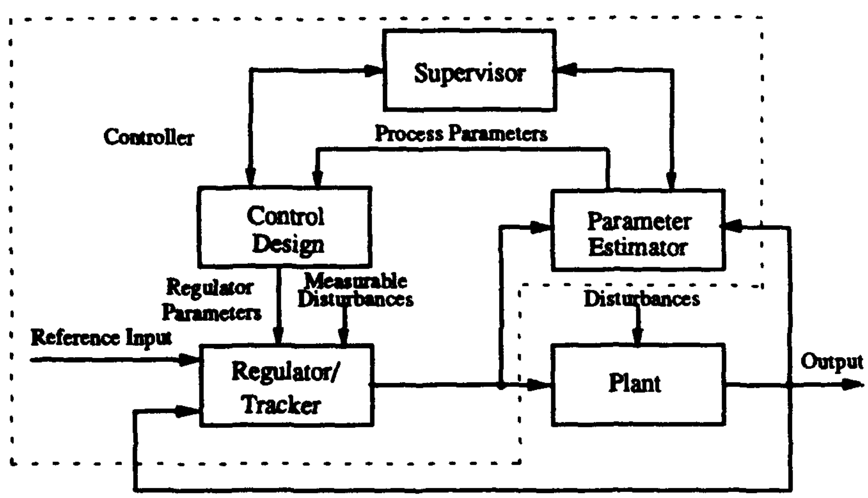


Figure 1. Diagram of a Self-Tuning Controller.

1. Regulator/tracker. This block is an algorithm that computes control signal on the basis of (a) feedback signal (feedback sensor output), (b) reference input (preplanned set point or time varying signal, or a priori unknown input signal), and (c) feedforward signal (feedforward sensor output).

2. Control design block. This block is conceptual. It includes (a) on-line regulator/tracker reparameterization that is taking place at every sampling period on the basis of current plant parameters computed by parameter estimator, and; (b) an off-line regulator/tracker design using an underlying controller synthesis method such as GPC and setting controller features, such as regulator structure, prediction and control horizons, and rate limits on control signal. However, if hardware permits, regulator/tracker redesign can be done in real time in closed loop to allow the on-line change of controller feature.

3. Parameter estimator. Within a specified model structure, this algorithm computes the parameters of the model to approximate plant input/output behavior on the basis of output measurements (feedback output sensor signal) and plant input (control signal).

4. Supervisor. This block includes: (a) the real-time diagnostics of the self-tuner, such as identifier conditioning, depending on the richness of online information in the input/output signals, (b) non-real-time operator/designer interface with an identifier for changing identification procedure and/or identifier structure, (c) non-real-time operator/designer interface with control design block for resetting "tuning knobs" in control design procedure and the regulator/tracker structure. If hardware permits, the last two tasks can be performed in real-time, resulting in an "intelligent" self-tuner.

The GPC Methodology

Design Components

The synthesis of the GPC regulator involves the following components:

1. A model that expresses the input/output plant behavior and typical plant disturbances in a finite set of parameters. The most suitable model structure for a large variety of plants is of the form:

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) &= b_0 u(t-1) + \cdots \\ &+ b_{n_b} u(t-n_b-1) + \alpha(t) + c_1 \alpha(-1) + \cdots + c_{n_c} \alpha(t-n_c) \end{aligned} \qquad \text{[Eq 6]}$$

and

$$\alpha(i) - \alpha(i-1) = \zeta(i) \qquad \text{[Eq 7]}$$

Where u(i), y(i), and $\alpha(i)$ are the values of plant input, output and disturbance at instant i, respectively. Sequence $\{\xi(i)\}^{\infty}_{-\infty}$ is an uncorrelated zero mean random pulse sequence with variance σ^2. Consequently from Equation 7 it is seen that $\{\alpha(i)\}^{\infty}_{-\infty}$, obtained by discrete integration from $\{\xi(i)\}^{\infty}_{-\infty}$ is a step sequence suitable for modeling the step-like disturbances. By defining q as a forward time shift operator:

$$qy(t) = y(t+1) \qquad \text{[Eq 8]}$$

and q^{-1} as a backward time-shift operator:

$$q^{-1}y(t) = y(t-1) \qquad \text{[Eq 9]}$$

and then by combining Equations 6 and 7 and factoring out y(t) and u(t-1), the so called Controlled Auto-Regressive Integrated Moving Average (CARIMA) model is obtained:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\frac{\xi(t)}{\Delta} \qquad \text{[Eq 10]}$$

where:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{n_b} q^{-n_b} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \\ \Delta &= 1 - q^{-1} (\textit{difference operator}). \end{aligned}$$

The model parameters are updated online by a parameter estimator that uses an a priori fixed model structure. The model structure (orders of polynomials $A(q^{-1})$, $B(q^{-1})$, and $C(q^{-1})$) can be changed on-line by the supervisor block or offline by the designer. The model primarily serves as a basis for the optimal prediction at time t of the future plant outputs y (t + j), j = 1, 2, ..., to the present and future controls.

2. A future reference sequence. {W (t + j)}, J = 1, 2, ... This might be either a fixed setpoint or preplanned sequence. However, the design procedure can be augmented to design the regulator/tracker for an a priori unknown reference sequence, as well.

3. A measure of the cost of the future predicted output errors {ê (t + j)} and the control effort {u (t + j – 1)}, is usually provided by a functional J ({ê (t + j)}, {u (t + j – 1)}). This measure denotes the optimal predicted value of y (t + j) given data up to time t by ŷ (t + j | t), and the future control increments (control "moves") by Δu (t + j) = u (t + j) – u (t + j – 1). In the GPC methodology, such a function, often simply referred to as a cost-function, is given by:

$$J_{GPC}(N_1,N_2,N_u,\lambda) = \sum_{j=N_1}^{N_2} [\hat{y}(t+j|t) - W(t+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(t+j-1)]^2 \qquad \text{[Eq 11]}$$

Where

ê (t + j) = ŷ (t + j | t) – W (t + j) is the predicted output error at instant t+j

N_1 is the cost-starting error horizon

N_2 is the output prediction horizon

N_u is the costing control horizon

λ is the control weighting.

The assignment of costing horizons (the boundaries of the time intervals where the cost is computed) for the output error and the input signal according to Equation 11 is done either offline by the designer, or online by a supervisor block. The quadruple [N_1, N_2, N_u, λ] can be viewed as a set of "tuning knobs" for a GPC algorithm.

4. An optimization procedure that at every current time moment t selects the future control sequence $\{u(t+j-1)\}$, $j = 1, 2, \ldots$, to minimize the cost function J_{GPC}. Control sequence can be determined, therefore, not further than the output prediction horizon, i.e., $\max\{j\} = N_2$. However, a simplified assumption is often made after the $(t + N_u - 1)$ steps-control increments are zero, which is equivalent to assigning an infinite cost to control "moves" after the $(t + N_u - 1)$ steps. If an optimization procedure is unconstrained, then it is possible to derive an explicit expression for the analytical calculation of the optimal control sequence. This allows a real-time implementation of GPC. Explicit expression can also be derived for some cases of constrained optimization, such as the rate limitation on the control signal, using the Lagrange multiplier techniques (Wilde and Beighter 1967).

5. Signal conditioning filters increase robustness of the algorithm against unmodeled dynamics and disturbances. In fact, the polynomial $C(q^{-1})$ in the plant model can be viewed not as a disturbance model, but as a filter (often noted as $T(q^{-1})$ polynomial) that must have roll-off at high frequencies to reduce predicted error due to unmodeled dynamics that might not be captured well at high frequencies by Controlled Auto-Regressive and Integrated Moving Average (CARIMA) models. Also, Δ can be viewed as an integrator that eliminates prediction error due to the model's inaccurate DC gain, and removes the effect of DC load disturbance on the output error.

Formulation of the Optimal Predictor

The key element in the GPC strategy is the optimal j-step-ahead predictor, an algorithm that uses all available input/output measurements up to and including time t, $\{y(t), y(t-1), \ldots, y(0), \Delta u(t-1), \Delta u(t-2), \ldots, \Delta u(0)\}$, as well as the future control increments, $\{\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+j-1)\}$, for computing the future output estimate $\hat{y}(t+j \mid t)$, which minimizes the variance of the j-step-ahead prediction error, $e_y(t+j \mid t) = y(t+j) - \hat{y}(t+j \mid t)$, where $y(t+j)$ is the true future j-step-ahead output.

Below, the j-step-ahead predictor is derived for the simplified case of $n_c=0$, i.e., $C(q^{-1})=1$. In this case, the CARIMA model takes the form

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{1}{\Delta}\xi(t) \qquad \text{[Eq 12]}$$

The derivation for $n_c>0$ is given in Favier (1987). Following Ljung (1991), the Diophantine identity:

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j^1(q^{-1}) \qquad \text{[Eq 13]}$$

is solved for $E_j(q^{-1})$ and $F_j^1(q^{-1})$, where $E_j(q^{-1})$ and $F_j^1(q^{-1})$ are polynomials of degree j-1 and n_a-1, respectively, which are uniquely defined given $A(q^{-1})$ and the prediction interval j. Multiplying Equation 12 by $q^j E_j(q^{-1}) \Delta$ yields

$$E_j(q^{-1})A(q^{-1})\Delta y(t+j) = E_j(q^{-1})B(q^{-1})\Delta u(t+j-1) + E_j(q^{-1})\xi(t+j) \qquad \text{[Eq 14]}$$

Rewriting Equation 13 as

$$E_j(q^{-1})A(q^{-1})\Delta = 1 - q^{-j}F_j^1(q^{-1}) \qquad \text{[Eq 15]}$$

and substituting Equation 15 into the left hand side of Equation 14 yields:

$$[1-q^{-j}F_j^{1}(q^{-1})] = E_j(q^{-1})B(q^{-1})\Delta u(t+j-1)+E_j(q^{-1})\xi(t+j) \quad \text{[Eq 16]}$$

Defining $G_j(q^{-1})=E_j(q^{-1})B(q^{-1})$, Equation 16 can be rewritten as

$$y(t+j) = G_j(q^{-1})\Delta u(t+j-1) + F_j^{1}(q^{-1})y(t) + E_j(q^{-1})\xi(t+j) \quad \text{[Eq 17]}$$

Substituting q=1 into Equation 13 yields $F_j^1=1$, $F_j^1(q^{-1})$, which can be factored as

$$F_j^{1}(q^{-1}) = 1+F_j(q^{-1})\Delta \quad \text{[Eq 18]}$$

where $F_j^1(q^{-1})$ is a polynomial in q^{-1}. Substituting Equation 18 into Equation 17 yields

$$y(t+j) = G_j(q^{-1})\Delta u(t+j-1) + y(t) + F_j(q^{-1})\Delta y(t) + E_j(q^{-1})\xi(t+j) \quad \text{[Eq 19]}$$

Since $E_j(q^{-1})$ is a polynomial of degree j-1, the last term in Equation 19 is unpredictable; therefore the best estimate $\hat{y}(t+j \mid t)$ of $y(t+j)$, given current and past output and control values as well as future control increments $\Delta u(t+j-1)$, $i \geq 1$, is obtained by dropping the last term in Equation 19:

$$\hat{y}(t+j \mid t) = G_j(q^{-1})\Delta u(t+j-1) + y(t) + F_j(q^{-1})\Delta y(t) \quad \text{[Eq 20]}$$

This is the so-called incremental predictor, which predicts the full value of y(t+j) using the input and output increments, $\Delta u(i)$ and $\Delta y(i)$. Denoting the coefficients of polynomials $G_j(q^{-1})$ and $F_j(q^{-1})$ as g_{ji}, $i=0,1, \ldots, n_g$, and f_{ji}, $i=0, 1, \ldots, n_a-1$, respectively, Equation 20 can be rewritten as:

$$\hat{y}(t+j \mid t) = \sum_{i=0}^{j} g_{ji}\,\Delta u(t+j-i)+p_j \quad \text{[Eq 21]}$$

where p_j is the signal whose components are known at time t:

$$p_j = \sum_{i=j+1}^{N_g} g_{ji}\Delta(t+j-i) + y(t)\sum_{i=0}^{N_a-1} f_{ji}\Delta y(t-i) \quad \text{[Eq 22]}$$

Thus, the prediction is split into two distinct parts: (1) p_j, the prediction of the output assuming no future changes in the control signal, in which case:

$$\sum_{i=0}^{j} g_{ji}\,\Delta u(t+j-i) = 0 \quad \text{[Eq 23]}$$

which corresponds to the predicted free future response; and (2) the predicted forced future response:

$$\sum_{i=0}^{j} g_{ji}\, \Delta u(t+j-i)$$

which is the convolution of the future control increments with plant step response, since the first j coefficients of $G_j(q^{-1})$ are just ordinates of the plant step response. Combining all the predictions $\{\hat{y}(t+j \mid t), j=1, \ldots, N_2\}$ into a vector $\hat{y}$ obtains the key equation:

$$\hat{Y} = G\tilde{u} + P \qquad \text{[Eq 24]}$$

where

$$\hat{Y} = [\hat{y}(t+1 \mid t),\ \hat{y}(t+2 \mid t),\ \cdots,\ \hat{y}(t+N_2 \mid t)]^T$$

$$\tilde{U} = [\Delta u(t),\ \Delta u(t+1),\ \cdots,\ \Delta u(t+N_2-1)]^T$$

$$P = [p_1,\ p_2,\ \cdots,\ p_{N_2}]^T$$

$$G = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ g_{N_1} & g_{N_2} & \cdots & g_1 \end{bmatrix}$$

Equation 24 is the basis for the derivation of the GPC control algorithm.

Formulation of the GPC Control Law

Figure 2 shows the conceptual framework for the derivation of the GPC control law. The derivation starts with obtaining a sequence of future control increments $\Delta u(t+j-1)$, $j=1, \ldots, N_u$, which minimize a cost function (Equation 11). By taking into account costing GPC horizons, the components of Equation 24 are given by:

$$\hat{Y} = [\hat{y}(t+N_1 \mid t),\ \hat{y}(t+N_1+1 \mid t),\ \cdots,\ \hat{y}(t+N_2 \mid t)]^T$$

$$\tilde{U} = [\Delta u(t),\ \Delta u(t+1),\ \cdots,\ \Delta u(t+N_u-1)]^T$$

$$P = [p_{N_1},\ p_{N-1+1},\ \cdots,\ p_{N_2}]^T$$

$$G = \begin{bmatrix} g_{N_1} & g_{N_1-1} & g_{N_1-2} & \cdots & 0 \\ g_{N_1+1} & g_{N_1} & g_{N_1-1} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{N_2-1} & g_{N_2-2} & g_{N_2-3} & \cdots & g_{N_2-N_u} \\ g_{N_2} & g_{N_2-1} & g_{N_2-2} & \cdots & g_{N_2-N_u+1} \end{bmatrix}$$

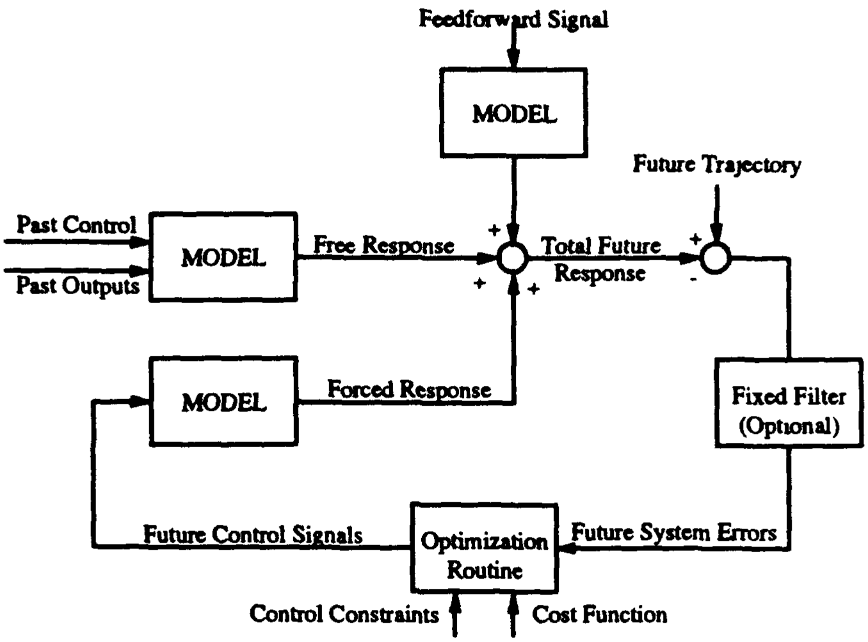


Figure 2. A Conceptual Framework of the GPC Law.

Introducing vector

$$W = [W(t+N_1),\ W(t+N_1+1),\ \cdots,\ W(t+N_2)]^T \qquad \text{[Eq 25]}$$

Equation 11 can be rewritten in vector-matrix form as

$$J_{GPC} = (W-\hat{Y})^T(W-\hat{Y}) + \lambda\tilde{U}^T\tilde{U} = (W-G\tilde{U}-P)^T(W-G\tilde{U}-P) + \lambda\tilde{U}^T\tilde{U} \qquad \text{[Eq 26]}$$

$\tilde{U}_{opt}$, which provides the global minimum of quadratic function Equation 26, is found by solving:

$$\frac{\partial J_{GPC}}{\partial \tilde{U}} = 0 \qquad \text{[Eq 27]}$$

which yields a computation formula for a single step of the GPC control law:

$$\tilde{U}_{opt} = (G^TG + \lambda I)^{-1}G^T(W-P) \qquad \text{[Eq 28]}$$

The GPC control law is generated using the "receding horizon strategy." At each sample-instant t,

1. Updated model parameters are passed from parameter estimator to update the parameters of the predictor and the single-step control algorithm. (In the non-self-tuning GPC version, this step is omitted.)

2. From the current measured output y(t) and the previous values of inputs and outputs, vector **P** of the predicted free plant response is computed and vector **W** of the future reference input sequence is evaluated

3. The future optimal control vector $\tilde{U}_{opt}$ is computed

4. Only the first element $\Delta u(t)$ of vector $\tilde{U}_{opt}$ is used to generate and apply the actual control signal $u(t)=u(t-1)+\Delta(t)$

5. At the next time moment, $t_{old}+1=t_{new}$, the whole procedure is repeated to account for a new output measurement $y(t_{new})$ and to compute $u(t_{new})=u(t_{new}-1)+\Delta u(t_{new})$.

Presently, efficient recursive algorithms can compute **P** via the multistep predictor formulation and the inversion of matrix $\mathbf{Q=G^TG+\lambda I}$ (Favier 1987). Multivariable GPC algorithms have also been developed (Lambert 1987; Favier 1987).

Recursive Least Squares (RLS)

Considering CARIMA model (Equation 10), which can be rewritten as an incremental model:

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t) + C(q^{-1})\xi(t) \qquad \text{[Eq 29]}$$

where:

$$\Delta y(t) = y(t) - y(t-1)$$
$$\Delta u(t-k) = u(t-k) - u(t-k-1)$$

If Equation 29 is put into a form that will fit into a Recursive Least Square (RLS) (Favier 1987; Ljung 1987; Ljung and Åström 1987):

$$y(t) = \theta'\phi(t) + \xi(t) \qquad \text{[Eq 30]}$$

where:

$$\theta = [a_1, a_2, \cdots, b_0, b_1, \cdots, c_0, c_1, \cdots]$$

composes of polynomials A, B, and C coefficients;

$$\phi(t) = [-\Delta y(t-1), -\Delta y(t-2), \cdots, \Delta u(t-1), \Delta u(t-2), \cdots, w(t-1), w(t-2), \cdots]$$

composes of input-output measurements;

$w(t)$ is an estimate of ξ.

RLS is based on the predicted error method. The RLS algorithm is the result of the minimization of the following cost function with respect to θ (Warwick 1988):

$$J = \frac{1}{2}\sum_{i=1}^{t} \lambda^{t-i} e^2(i) \qquad \text{[Eq 31]}$$

where

e(t) is the estimated error between the current output (y(t)) and the estimated output ($\hat{y}(t)$)

λ is the forgetting factor.

Equation 31 can be rearranged as:

$$J = \frac{1}{2}\sum_{i=1}^{t}\lambda^{t-i}(y(t)-\theta'\phi(t))^2 \qquad \text{[Eq 32]}$$

The forgetting factor λ is a data weighting factor that weights the current data more than the old data. That is to say, it forgets the old data. This is an ad-hoc method for dealing with uncertain estimation. Finally, minimization of Equation 33 gives (Warwick 1988):

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t)(\Delta y(t) - \hat{\theta}'(t-1)\phi(t)) \qquad \text{[Eq 33]}$$

$$P^{-1}(t) = P^{-1}(t-1) + \phi(t)\phi'(t) \qquad \text{[Eq 34]}$$

where, **P(t)** is the covariance matrix. The covariance matrix **P(t)** is a measure of the convergence of the estimated parameters. In general, if the trace of **P(t)** is small, generally less than one, the estimated parameters will approximate the true parameters. If the trace of **P(t)** is large, the estimated parameters are uncertain.

The computation of Equation 34 and Equation 35 is expensive, since at each sampling interval, a matrix inversion is performed. However, the matrix inversion lemma presented in Åström and Wittenmark+ (1989) can be employed to reduce the computing time and improve numerical accuracy, from the lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad \text{[Eq 35]}$$

where:

A, C, and (C^{-1} +D A^{-1} B) are nonsingular square matrices. That is, inversion is substituted by multiplication and addition. The matrix inversion lemma is next applied to Equation 35. This leads to the standard RLS (Warwick 1988):

estimated error,

$$e(t) = \Delta y(t) - \hat{\theta}(t-1)\phi(t) \qquad \text{[Eq 36]}$$

Kalman gain update,

$$K(t) = \frac{P(t-1)\phi(t)}{[\lambda + \phi'(t)P(t-1)\phi(t)]} \qquad \text{[Eq 37]}$$

Covariance matrix update,

$$P(t) = [P(t-1) - \frac{p(t-1)\phi(t)\phi'(t)P(t-1)}{\lambda + \phi'(t)P(t-1)\phi(t)}]\frac{1}{\lambda} \qquad \text{[Eq 38]}$$

estimated parameters update,

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)e(t) \qquad \text{[Eq 39]}$$

That is,

$$[\text{new estimates}] = [\text{old estimates}] + \text{Kalman gain} * [\text{estimated errors}] \qquad \text{[Eq 40]}$$

Equation 38 is not numerically well-conditioned. A better way to update the covariance matrix **P(t)** is to update the square root of **P** instead of **P**. Rewrite **P = UDU'**, where **U** is the upper triangular matrix of **P** and **D** is the diagonal matrix of **P**. The square root of **P** becomes $UD^{-1/2}$. This procedure is referred to as UD factorization (Bierman 1977).

In general, the RLS algorithm does not estimate the C (color noise) polynomial. An extension of the RLS algorithm will approximate the random noise term. This is called extended RLS (ELS) (Ljung and Åström 1987). It is impossible to estimate random white or color noise directly. However, introducing another variable, w, obtains (Ljung and Åström 1987):

$$w(t) = \Delta y(t) - \hat{\theta}'(t-1)\phi(t) \qquad \text{[Eq 41]}$$

Equation 38 will give the approximate noise at t. That is, w(t) will be approximately equal to $\xi(t)$. For simplicity, RLS and ELS will be considered synonymous for the balance of this report.

The RLS algorithm cannot track slow time-varying parameters because when the estimated error becomes small, so do the covariance matrix **P** and Kalman gain **K**. This is one of the major drawbacks of RLS, which does not retain alertness or adaptivity. When the parameters drift (i.e., when the estimated error is large), the **P** matrix and **K** do not change. Therefore, the RLS algorithm gives poor and sometimes wrong estimates for time-varying systems. However, there is a parameter available in the RLS algorithm that can be used to address this problem—the forgetting factor λ. Even though the forgetting factor can take on any value between zero and one, it has been found that 0.90 can be considered as a lower bound and that 0.95 is a good starting or default value in industrial practice (Warwick 1988). In general, the constant forgetting factor will improve the estimation of slow time-varying systems. However, at the steady state (i.e., where the parameters do not vary or vary very slowly) with disturbances, the RLS algorithm may cause the estimated parameters to oscillate. Therefore, the RLS with constant forgetting factor is not recommended for long-term identification.

There are many ad-hoc procedures to keep the RLS algorithm "awake," including leakage, constant trace, covariance resetting, dead zone, and others. However, these ad-hoc procedures require process knowledge to work. For example, the RLS with covariance resetting prevents the covariance **P** from falling below a certain level. If the covariance **P** is below that level, it is reset to some large value. The lower bound of **P** and the value to which **P** is reset are process-dependent. Human operators must enter the process information before the RLS algorithm starts. Such ad-hoc procedures are not very effective in developing a general purpose identification scheme. Another way to modify the RLS algorithm is to allow the forgetting factor to vary. The variable forgetting factor is changed according to the estimated error. If the estimated error is large, $\lambda(t)$ will be set to a small value. If the estimated error is small, $\lambda(t)$ will be set to one. By replacing Equations 39 and 40, the RLS with variable forgetting factor becomes (Warwick 1988):

$$\lambda(t) = 1 - \frac{[y(t)-\phi'(t)\hat{\theta}(t-1)]^2}{[1+\phi'(t)P(t\ 1)\phi(t)]\sigma} \quad \text{[Eq 42]}$$

$$K(t) = \frac{P(t-1)\phi(t)}{[1+\phi'(t)P(t-1)\phi(t)]} \quad \text{[Eq 43]}$$

$$W(t) = P(t-1)-K(t)\phi'(t)P(t-1) \quad \text{[Eq 44]}$$

where:

$\lambda(t) >= 0.90$ for practical implementation

$\sigma/\sigma_w \approx 1000$, σ_w is the variance of the zero mean noise sequence.

The above development assumed an input signal that is persistently excited, i.e., that the input signal contains all frequency modes. Under feedback control, there is no guarantee that the input signal is persistently excited. When the input signal is not persistently excited, Equation 39 becomes $\mathbf{P(t)=P(t-1)/\lambda}$. Therefore, the covariance matrix **P** will "blow up" if the input signal is not "rich enough." This phenomenon is usually referred to as estimator "windup" (Harris and Billings 1985). A detailed discussion will be presented in the self-tuning control section after the introduction of the generalized predictive control (GPC) law.

Inclusion of Constraints on the Control Signal in the GPC Control Law

While both magnitude and rate constraints on control signal might be needed in a specific implementation, it is numerically more efficient to introduce only rate constraints that usually indirectly ensure adequate control signal magnitudes as well. Thus, it is assumed that control "moves" must stay within the prescribed feasibility bound during the whole costing control interval:

$$\alpha_j \leq \Delta u(t+j-1) \leq \beta_j,\ j = 1, \cdots, N_u \quad \text{[Eq 45]}$$

To derive an algorithm for the computation of the constrained optimal control vector denoted as $\tilde{\mathbf{U}}_c$, assume, for simplicity, that only one control increment of $\tilde{\mathbf{U}}_c$ saturates, i.e., it needs to be on the boundary, for example:

$$\Delta u(t+j-1) = \alpha_j \quad \text{[Eq 46]}$$

In this case it is possible to find $\tilde{\mathbf{U}}_c$ using a Lagrange multiplier technique. For this purpose, function:

$$L = (G_1\tilde{U}_c+P-W)^T(G_1\tilde{U}_c+P-W) + \lambda\tilde{U}_c^T\tilde{U}_c + 2\mu_i[\Delta u(t+j-1)-\alpha_j] \quad \text{[Eq 47]}$$

is minimized by solving $\partial L/\partial \tilde{U}_c = 0$. This yields:

$$\tilde{U}_c = (G_1^T G_1 + \lambda I)^{-1} G_1^T (W-P) + (G_1^T G_1 + \lambda I)^{-1} \mu_j e_j \qquad \text{[Eq 48]}$$

and

$$e_j = [0, \cdots, 1, 0, 0]^T \qquad \text{[Eq 49]}$$

or

$$\tilde{U}_c = \tilde{U}_{opt} + (G_1^T G_1 + \lambda I)^{-1} \mu_j e_j \qquad \text{[Eq 50]}$$

Combining the jth equation of Equation 50 with Equation 46 yields

$$\alpha_j = \Delta U_{opt}(t+j-1) + h_{jj}\mu_j \qquad \text{[Eq 51]}$$

where h_{jj} is an element (j,j) of matrix $(G_1^T G_1 + \lambda I)^{-1}$. Equation 52 is solved for unknown μ_j, and finally $\tilde{U}_c$ is obtained from Equation 50. This procedure generalizes to any number of saturated controls in vector $\tilde{U}_c$ (Clarke and Tsang 1988). A heuristic choice of saturated elements of $\tilde{U}_c$ is to pick those elements of $\tilde{U}_{opt}$ that violate feasibility boundaries.

Commissioning of a Self-Tuning Regulator

At the beginning of the tuning phase, an adaptive control law should not be used because the process parameters are uncertain. Therefore, the self-tuning control (STC) requires a start-up procedure.

There are two ways to start-up the STC: open-loop identification and closed-loop identification (Mohtadi 1988). For the open-loop identification, the process parameters are estimated in open-loop fashion with the pseudo-random binary sequence (PRBS) as the input signal. In general, the commissioning period would be from about 50 to 100 samples. However, for industrial process control, the commissioning period may require several hundred samples due to the long time constants of the processes. After the commissioning period, the adaptive controller "kicks in" and the PRBS is cut off.

Closed-loop identification uses a proportional, proportional-integral, or proportional-integral-derivative controller to close the loop. The settings for the PID controller will be conservative. However, pretuning the PID controller is required (Appendix B). Also, an additional PRBS signal is added into the control signal to improve the estimation. After the commissioning period, the controller and the PRBS are disengaged. If the process is open-loop stable, both start-up procedures can be used. However, if the process is open-loop unstable, the closed-loop identification procedure must be used. After the commissioning period, the input signal may not be persistently excited. A good control law will try to make the control signal as smooth as possible. Unfortunately, identification needs a persistently excited plant to perform well. At the steady state, a probing signal (e.g., PRBS, periodic signal, etc.) is added to the control signal to ensure the performance of the estimator (McDonald, Kwatney, and Spare 1971). However, the probing signal may cause the output to ring. If the oscillation of output cannot be tolerated, the estimator must be turned off. STC always has conflicting objectives. The estimator performs well with persistent excitation, but good control dictates keeping the input signal as constant as possible. Therefore, there is a constant trade-off between identification and control requirements and objectives.

Program Descriptions

The Software Structure of the Control Algorithm

The control algorithm is a timer-initiated interrupt procedure operating on the µMac when the µMac is in control mode. The µMac interrupt system remains enabled during most of the algorithm to permit plant data scanning to proceed (continued scanning will permit the implementation of digital filtering of the input data stream at some future time). The interrupt system is disabled only while current plant data are copied to buffers used for the control computation. Copying the scanned data assures that the plant data used in the control computation will not be modified as the computation proceeds.

The sequence of the control algorithm follows. The data needed for the steam pressure loop is copied first, and then the firing rate to steam pressure controller (GPC) is executed. This yields a firing rate command, used to calculate fuel flow based on fuel energy content calculations. The resulting fuel flow is interlocked against available airflow (to prevent incomplete combustion) and then the desired fuel flow command is sent to the fuel pump. Next current plant data are again copied to the control buffer, and a control calculation for the excess oxygen loop is performed. This calculation results in a new command signal for the blower dampers. Finally, current plant data are copied one final time, the water level control is computed, and the water pump setting command is issued.

Figures 3 to 5 show the main control loops (outer loops) for drum pressure, oxygen level, and drum water level. Each of these main loops contains additional controllers around the actuators (fuel flow, air flow, and feedwater flow inner control loops) to assure that the command signal sent by the controller is the signal sent by the actuator. These inner loops are displayed in Figures 6 to 8. It is noted that all three main control loops (drum pressure, excess oxygen level, and drum water level) are computed using the same GPC function (with different parameters, of course). This structure permits easy substitution of alternate control strategies.

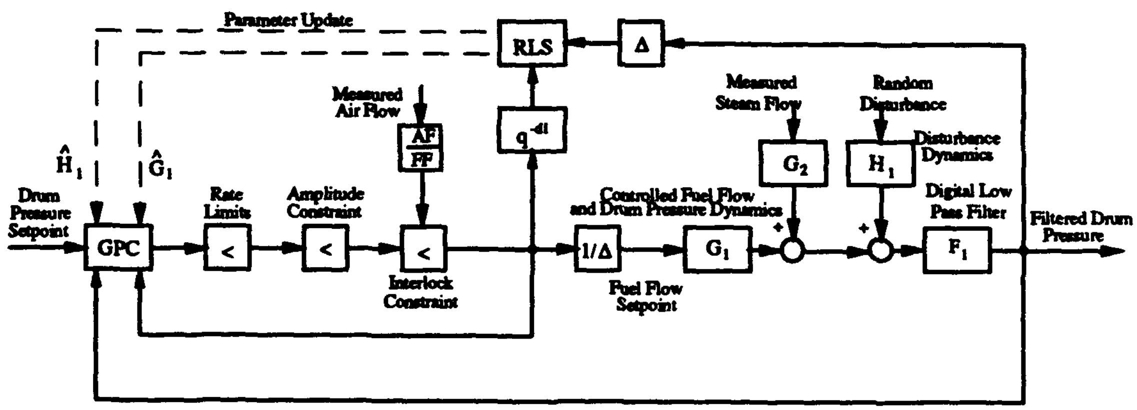


Figure 3. Drum Pressure Control Loop.

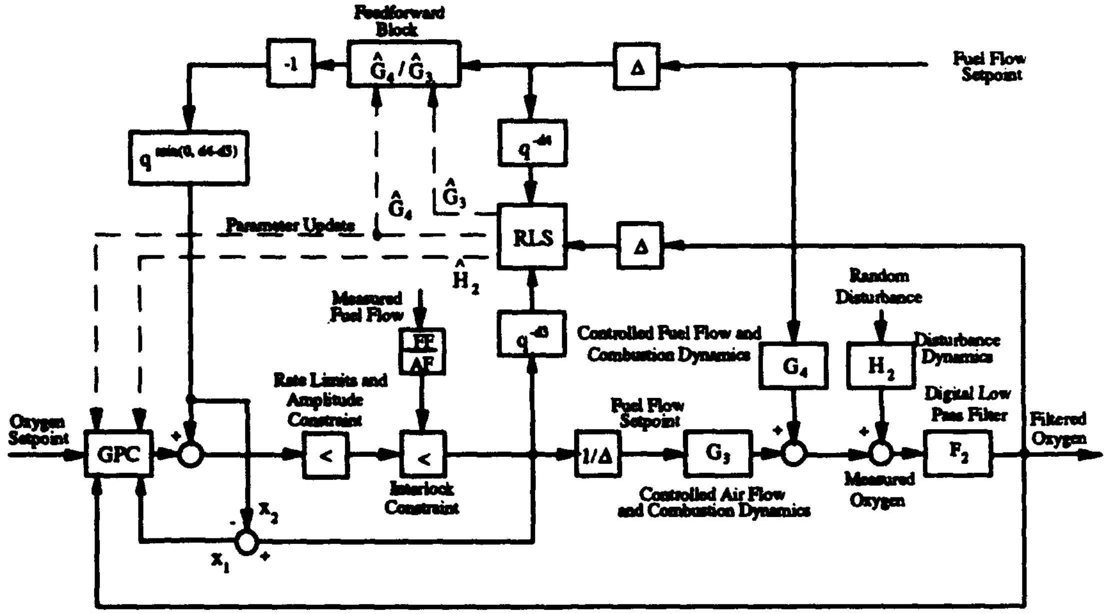


Figure 4. Oxygen Level Control Loop With Interlocks and Feedforward.

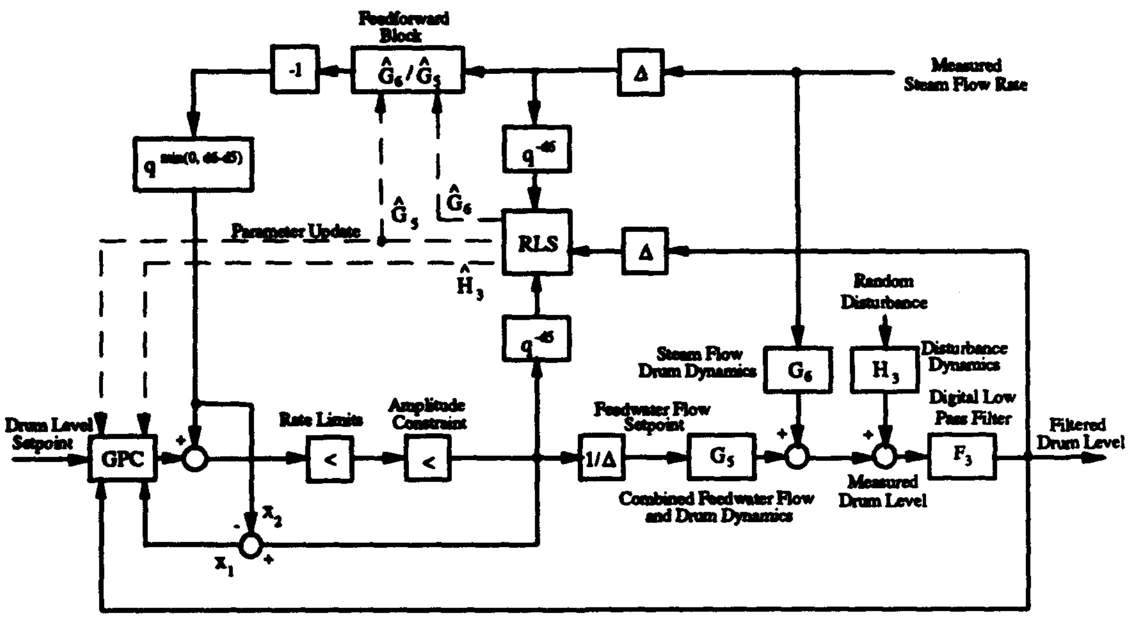


Figure 5. Water Level Control Loop With Feedforward Blocks.

Controller Structure and Logic

<u>Feed Forward Loops</u>. For a perfectly known model and perfectly measured disturbance, the output of the plant can be made precisely equal to the setpoint regardless of the type of disturbance, without the need for feedback control. However, even in our idealized model, there are other nonmeasured disturbances. For example, coupling between loops will produce an effect in subsystems indirectly affected by a particular input. Specifically, in the nonlinear model, the header pressure is indirectly affected by changes in water flow. Another example of nonmeasured disturbance is noise in sensors and actuators. Because of these nonmeasured disturbances, feedback control is still necessary. The use of feedforward improves the regulation since there is no wait for the error to show up in the output before taking corrective actions. Also the process of identifying plant parameters for self tuning is made simpler by the partial cancellation of disturbances.

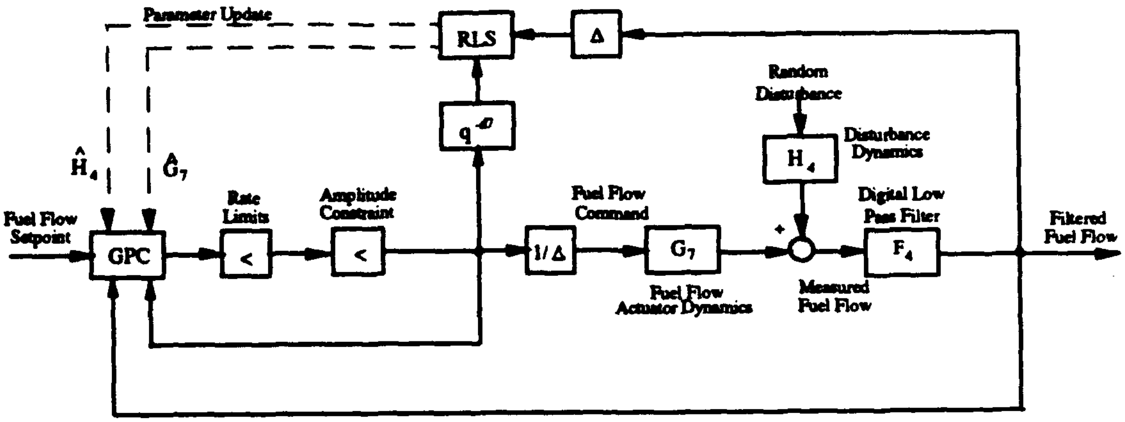


Figure 6. Fuel Flow Inner Control Loop.

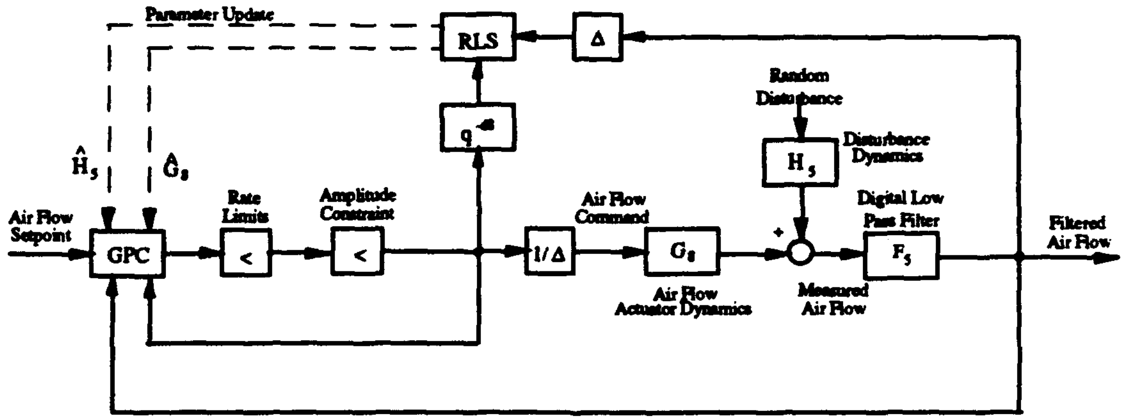


Figure 7. Air Flow Inner Control Loop.

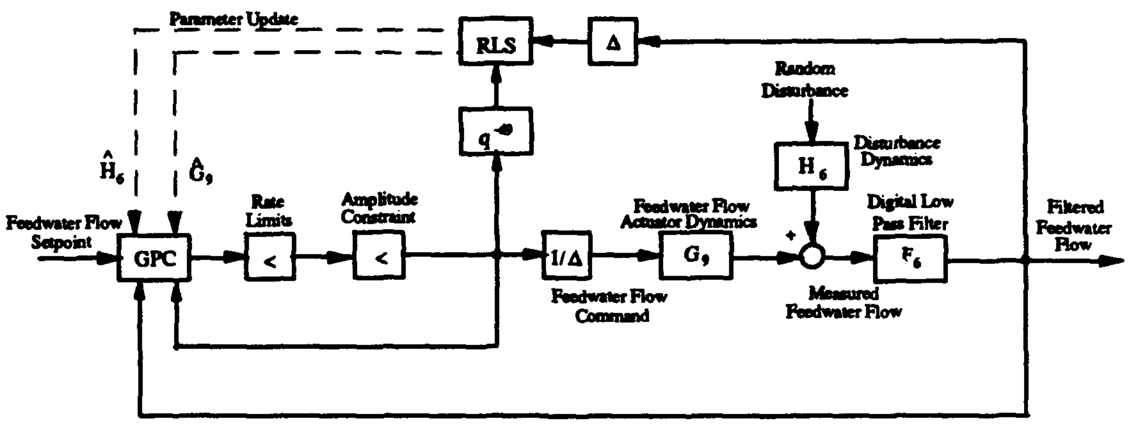


Figure 8. Feedwater Flow Inner Control Loop.

A feedforward controller was chosen for the fuel flow/header pressure loop to reject the effect of a varying "steam demand" on the system. Consider steam demand as the position of an imaginary valve in the output steam line. Steam demand is one of the inputs to the nonlinear model of Bell and Åström, and is retained as an input in both the linear coupled and decoupled models. The steam demand is not a measurable quantity, and therefore cannot be used as a feedforward disturbance. The outlet steam flow rate, however, can be measured, and the transfer functions from steam demand to steam flow and from steam demand to header pressure may be calculated from the linear model. By dividing these transfer functions, the overall transfer function from the measurable steam flow to the output header pressure is obtained. This transfer function is used in the feed forward controller design suggested by Stephanopolous (Wahlberg 1990).

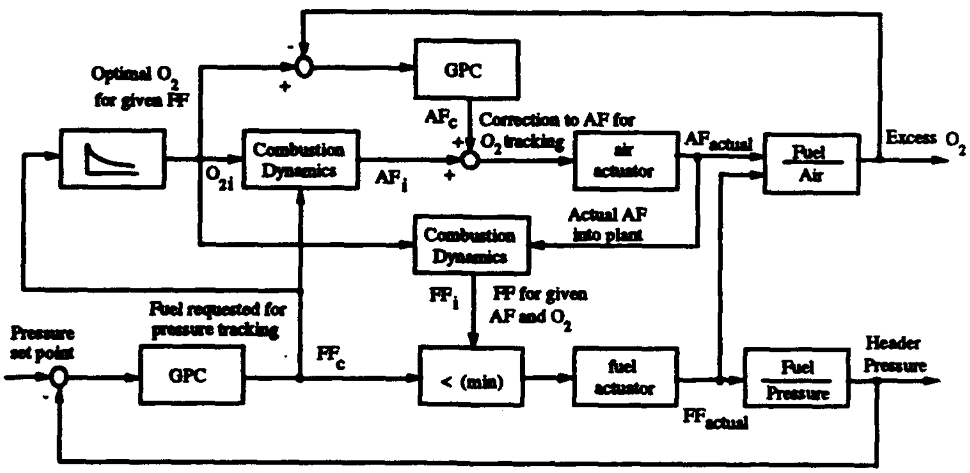
Attempts to directly apply the feedforward design techniques described above to the water flow/water level loop could possibly fail due to the non-minimum phase (i.e., shrink and swell) characteristics of the drum. When the feed forward design is carried out, an unstable pole appears in the feedforward transfer function. Since disturbance signals pass through this transfer function in the open loop, the controller cannot be stabilized by the plant and the control signal saturates. Alternative feedforward designs for controlling drum water level are currently under investigation.

Interlocks and Cross Limit Constraints. To ensure that excess oxygen is always maintained at or above the values required for safe operation, and if possible at optimum levels, a cross-limiting procedure was implemented to calculate fuel flow and air flow control inputs to the plant (Figure 9).

The excess oxygen setpoint is taken as the optimum excess oxygen for the current fuel flow conditions (Figure 10). The current low fuel conditions are obtained as the minimum of the fuel flow requested for tracking the header pressure and the fuel flow allowed by the current available air. The approximate air flow needed to match the fuel at the desired firing rate is calculated from combustion dynamics of the linear model.

The constraint on fuel flow ensures that even when load conditions change rapidly, there will always be sufficient excess oxygen in the combustion chamber to ensure safe operation (if necessary, at the expense of a drop in header pressure). Higher efficiency and lower pollutant emission are achieved through the proper selection of the excess oxygen set points, which are functions of the given boiler load.

One may question the need for feedback on the excess oxygen loop. It may appear that if the interlocks are properly designed, the current air flow for obtaining the optimal excess oxygen under the current fuel flow conditions will be asserted. The problem is that the interlocks rely on a linear combustion model that may not be valid outside a certain operating range. The purpose of feedback, then, is to eliminate any steady-state error in excess oxygen due to nonlinearities. However, the addition of



AF - Air Flow c - Controller value
FF - Fuel Flow i - Interlock value

Figure 9. Interlock (Cross Limit Constraint) Configuration.

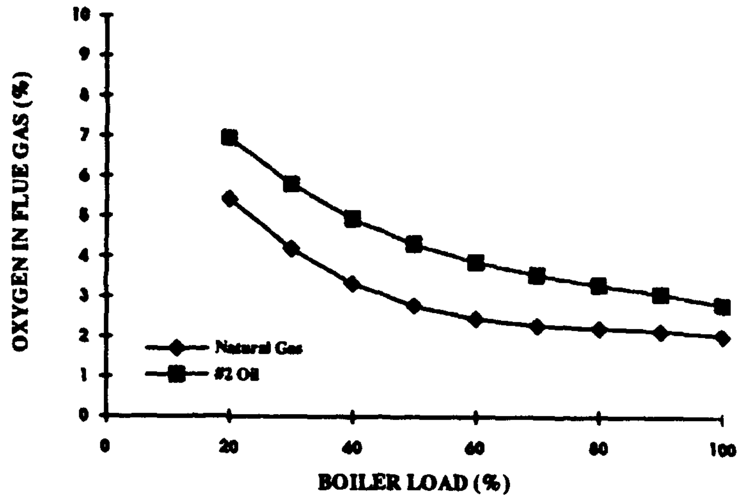


Figure 10. Optimal Excess Oxygen for a Given Load.

is to eliminate any steady-state error in excess oxygen due to nonlinearities. However, the addition of feedback on the excess oxygen loop may also introduce dangerous transients into the air flow signal, defeating the purpose of the interlocks.

Probably the best solution to this problem is to use feedback with the interlocks, but to place constraints or heavy weights on the control signal. This may be performed by using either a GPC or PID controller and should eliminate both the steady-state error and the risk of bad transients. Another solution is to use recursive least squares identification to provide the interlocks with a more reliable combustion model. However, this approach may place excessive reliance on the performance of the identification. Further testing with various boiler models will be necessary before a definite conclusion can be reached.

Control System Flow Chart

Figure 11 shows the main program flow chart. Figure 12 shows the flow chart for the µMAC-6000 controller data-processing procedure, and Figures 13 and 14 show the flow charts for inner and outer loops, respectively.

Startup and Shutdown Procedures

The automatic startup and shutdown procedures were implemented using the FIX DMACS software package running on a 386SX PC. FIX DMACS* is multitasking software made up of foreground tasks and background tasks that provide complete monitoring and control over automated processes. The foreground tasks are programs that operators interact with to set up the system, display data, and perform other functions. The background tasks, such as Schedule Reports and Tasks, Alarm Monitoring and Processing, once started, run without intervention from the operator. The FIX DMACS package has been installed on the master control console of the prototype system. The database code for automatic startup and shutdown procedures are presented in Appendix C.

*FIX refers to Fully Integrated Control System; DMACS refers to Distributed Manufacturing, Automation, and Control Software.

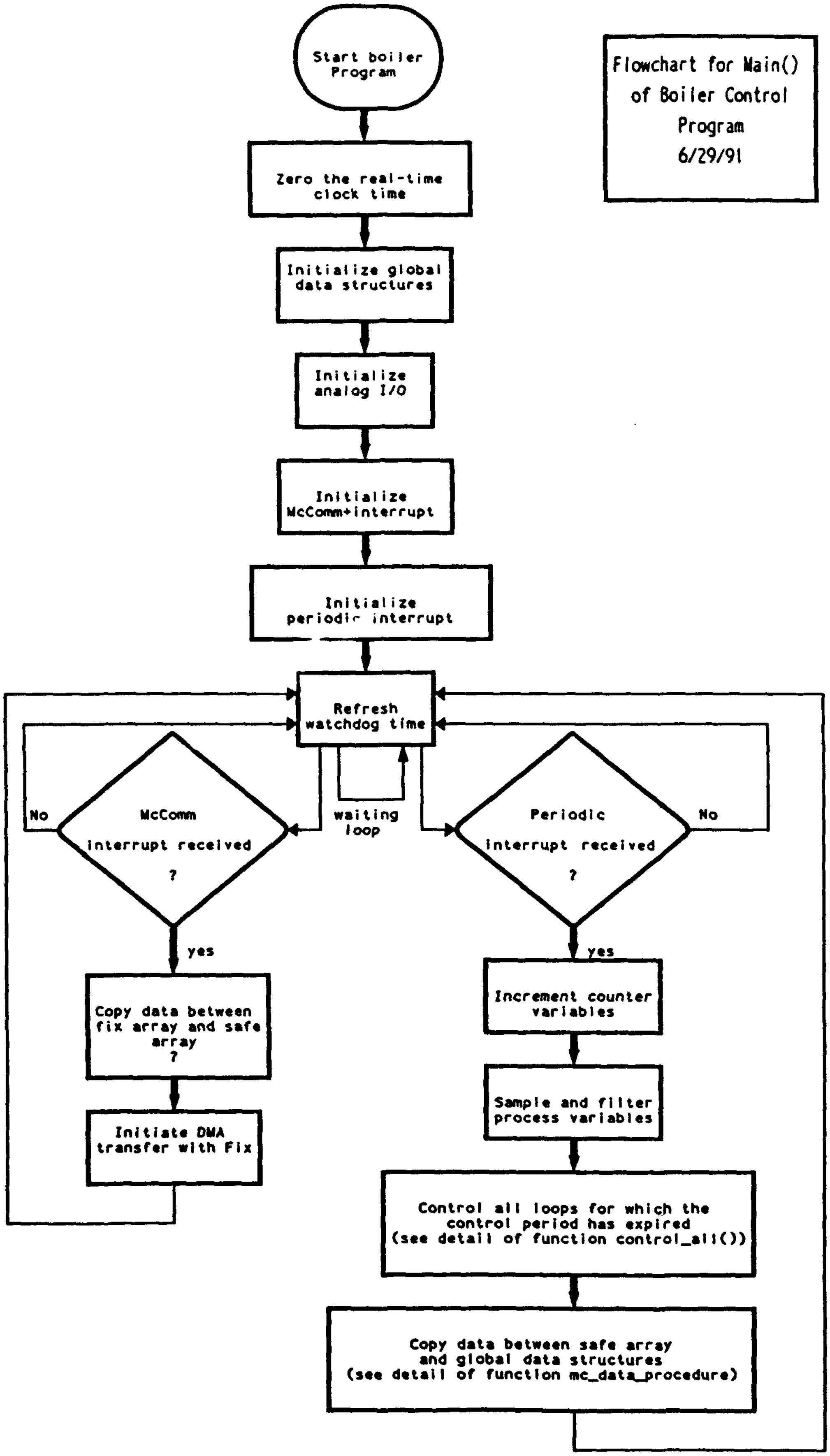


Figure 11. The Flow Chart of the Main Program.

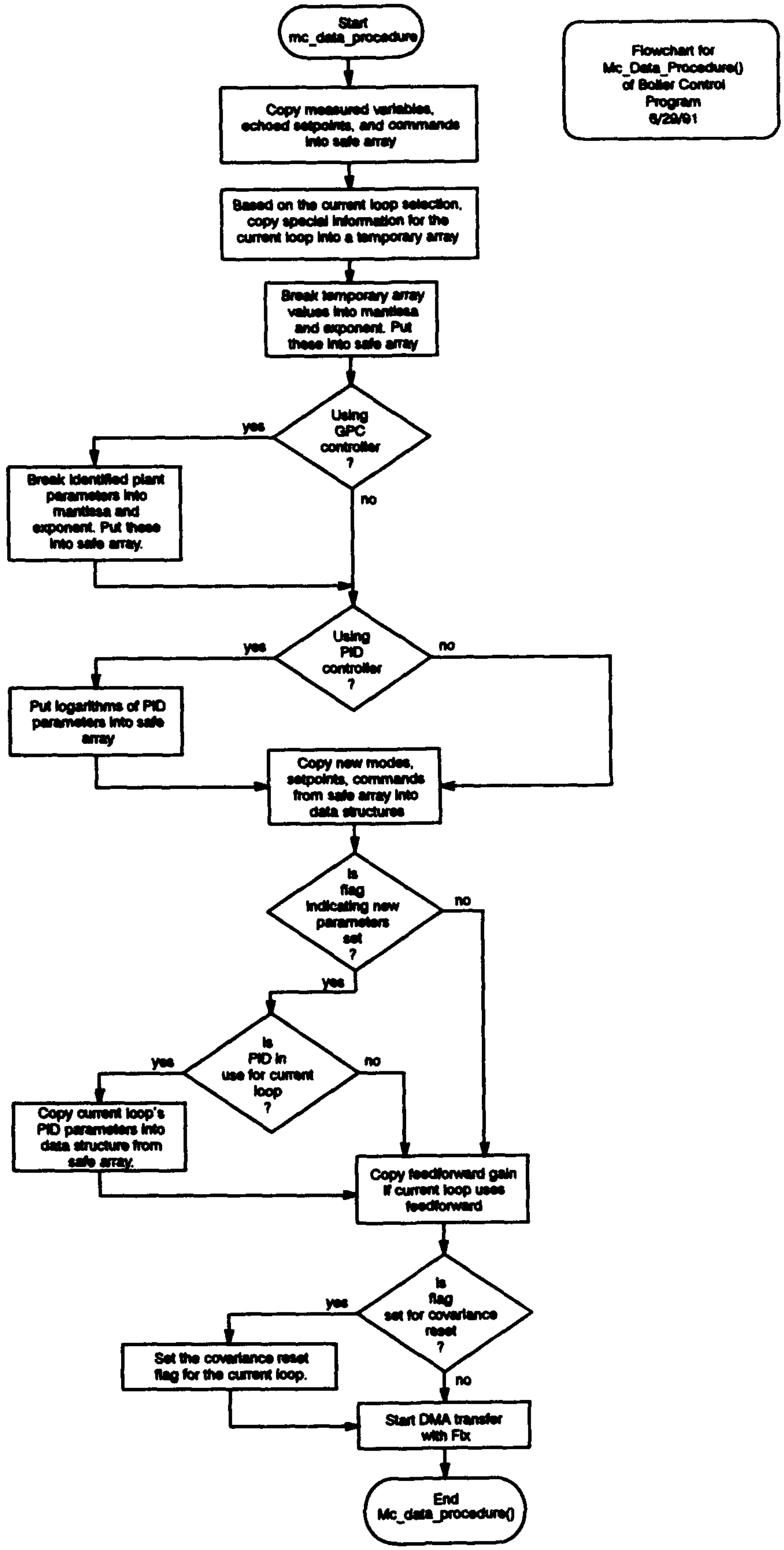


Figure 12. Flow Chart for the MAC Data Processing Procedure.

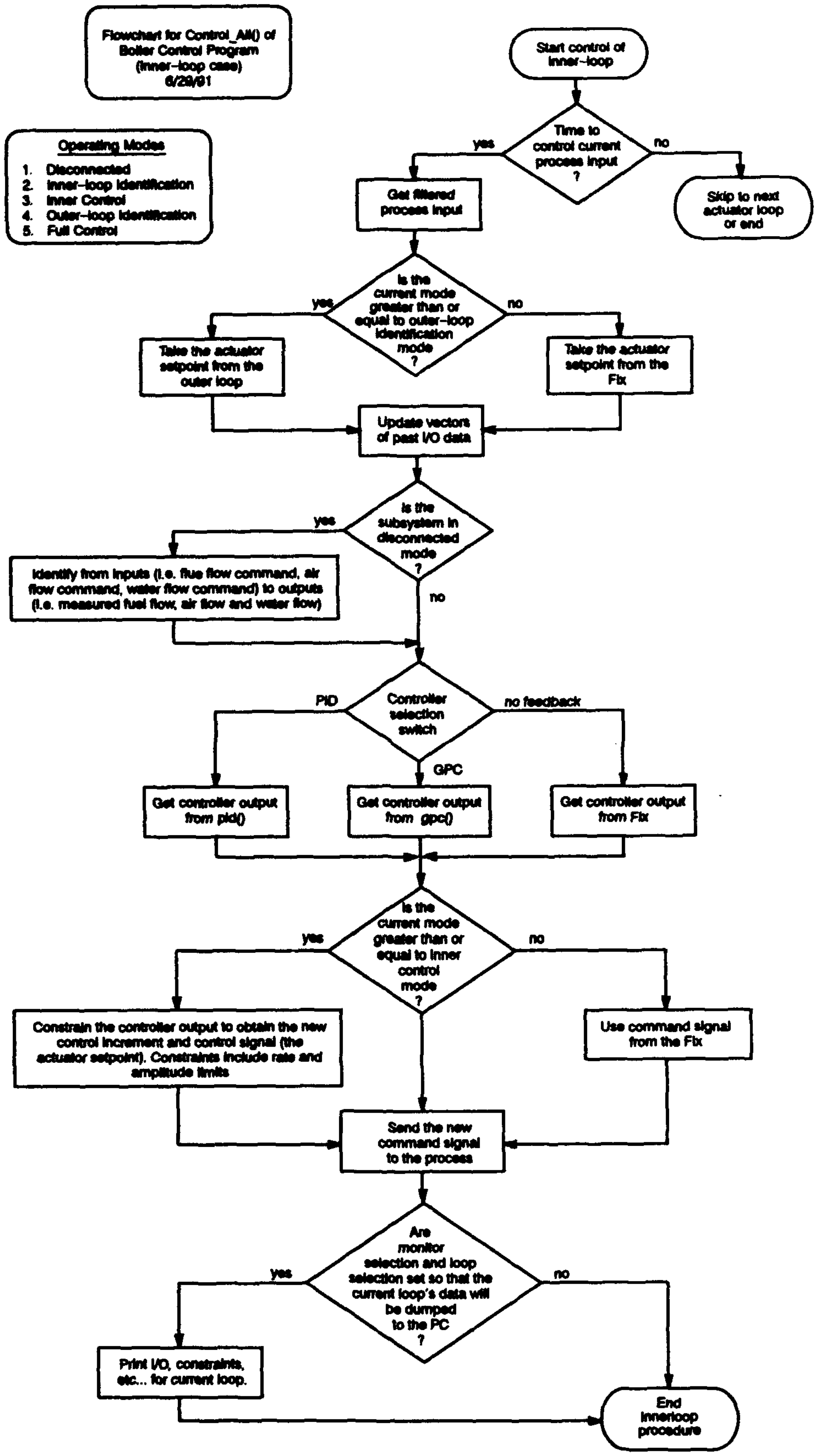


Figure 13. The Flow Chart of the Inner Loop Control Procedure.

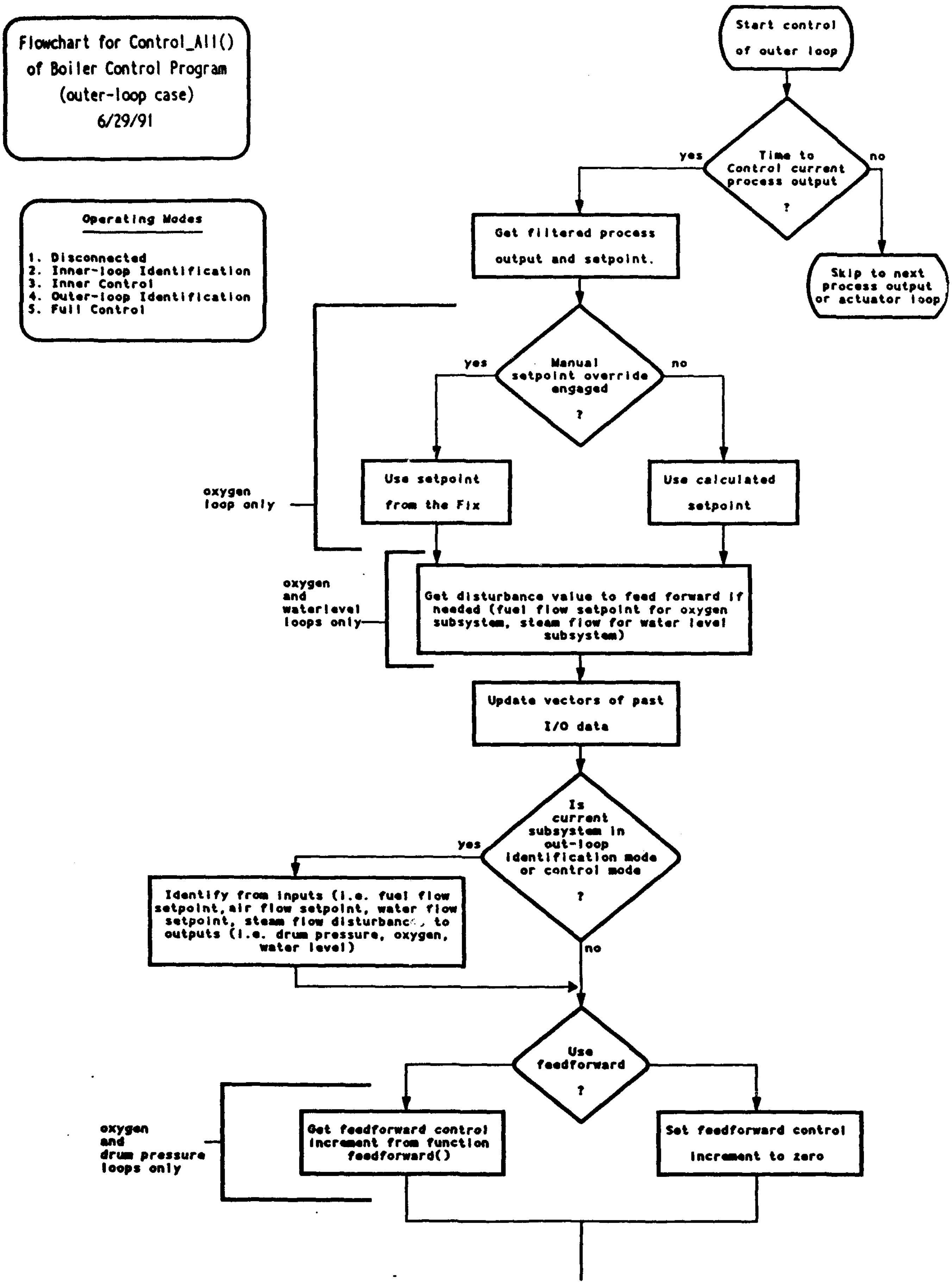


Figure 14. The Flow Chart of the Outer Loop Control Procedure.

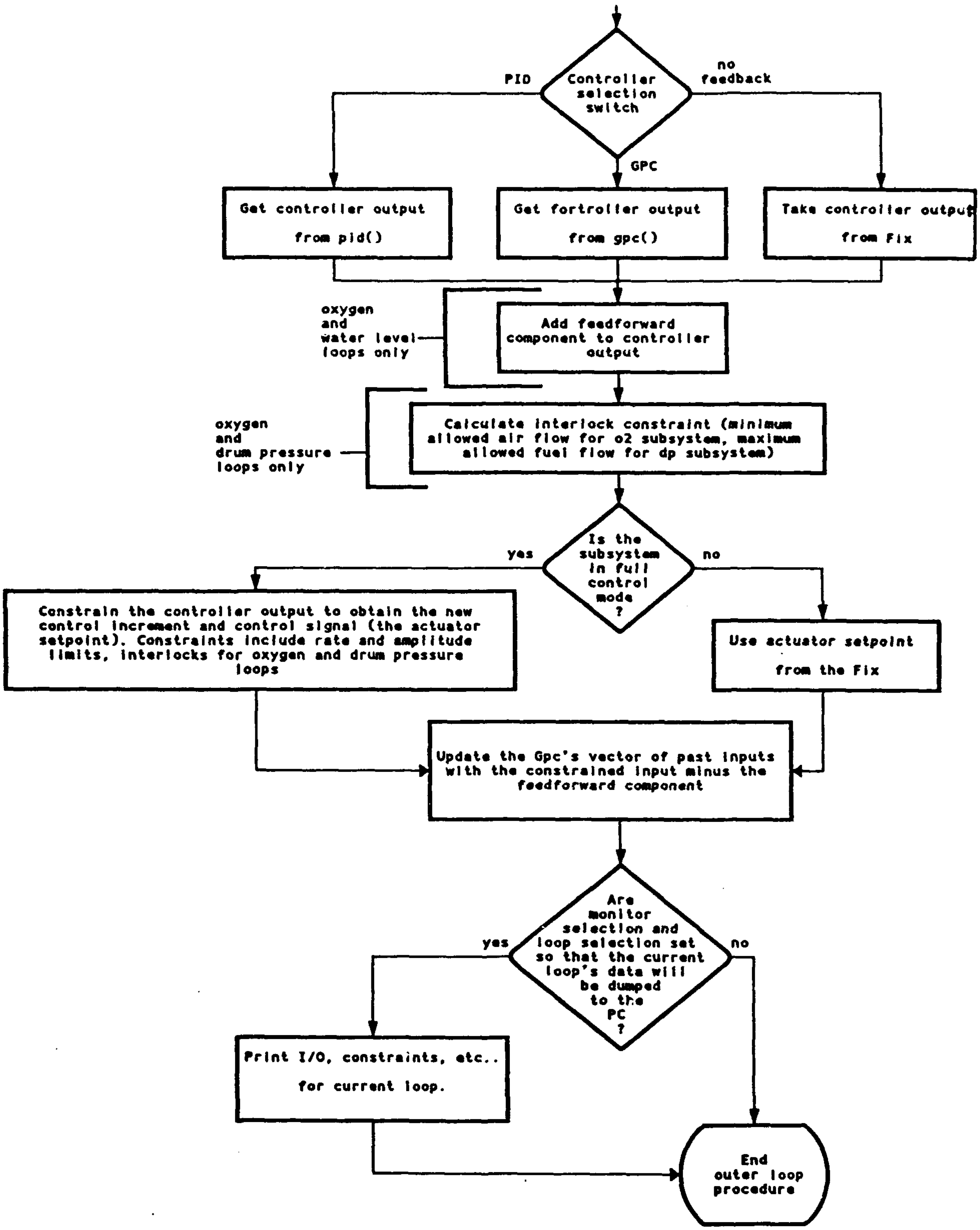


Figure 14. (Cont'd)

The startup and shutdown procedures detailed below were developed in accordance with NFPA 85B "Standard for Explosions in Natural Gas-Fired Multiple Burner Boiler-Furnaces" (National Fire Protection Association 1989). Observations made at the Abbott Power Plant were also considered.

Startup Procedure

The startup sequence will comply with NFPA 85B section 5-2.1.2, and will have to be customized for every plant based on boiler manufacturer's recommendations. The sequence is implemented using the FIX DMACS software, and can easily be reconfigured to suit any boiler. A generalized startup sequence is:

1. Conduct a prestartup check per NFPA 85B section 5-2.1.1. In particular, bring the drum water level to the normal operating point and engage the drum water level controller. Open dampers and verify an open flow path from air inlets to the stack. Prompt operator if manual checks are required.

2. Start fans and achieve a purge air flow rate in accordance with NFPA 85G. Engage and use the air flow control loop to establish the air flow rate.

3. If the boiler is equipped with regenerative type air heaters and gas recirculating fans, start them, or prompt the operator to start them.

4. Precharge the burner header with gas per NFPA 85B section 5-2.1.2 item d and check for leaks.

5. Perform a unit purge.

6. Close the main fuel control valve, then open the main safety shutoff valve.

7. Establish correct header pressure for burner lightoff.

8. Establish correct igniter header pressure for ignition.

9. Adjust the air register or damper on the burner selected for light-off to the correct light-off position.

10. Attempt to ignite the first igniter. If a flame is not established on the igniter within 10 seconds, close the igniter safety valve and determine and correct the cause of the ignition failure. Wait at least 1 minute before a retry. Halt the startup sequence after the second failed attempt to ignite the igniter.

11. Note that variations of the procedure are required for boilers equipped with Class 3 special electric igniters.

12. Ensure that the igniter is operating correctly and proceed to supply fuel to the main burner. Failure to ignite within 5 seconds will cause a master fuel trip. After a trip, the startup sequence will be repeated from step 6. The second failure to lightoff will halt the startup procedure.

13. Wait for operator verification that a stable flame has been established. Ramp burner air registers or dampers to the normal operating position. Engage the fuel flow control loop with setpoint corresponding to the current fuel flow level. Be sure that ignition is not lost.

14. Shut down the igniters and verify that a flame is maintained.

15. Shut off the burner header atmospheric valve if necessary (see NFPA 85B section 5-2.1.2 item p).

16. If the boiler is equipped with other burners, place them in service as required and as specified in NFPA 85B section 5-2.1.2 item q.

17. For multiple burner systems, ignite the maximum number of burners consistent with anticipated continuous load.

18. Verify that a stable flame has been established. Ramp fuel flow to the minimum operating level if necessary. Nonoperating burners should be correctly placed out of service. Ramp airflow down to a level such that the excess oxygen level is about 10 percent above the normal operating target. Using this setting as the excess oxygen setpoint, engage the excess oxygen control loop. Verify burner operation again. Engage automatic excess oxygen setpoint control. Verify burner operation. Use drum pressure as the pressure loop controlled variable and close the steam pressure loop. Ramp up the drum pressure to match the steam header pressure. Switch the steam pressure controlled variable to header pressure to place the boiler on line.

Shutdown Procedure

The shutdown procedure complies with NFPA 85B section 5-2.3. The following generalized procedure will be customized as required for individual installations:

1. Switch the pressure loop controlled variable to drum pressure to bring the boiler off line.

2. Disengage the pressure loop.

3. Ramp down the fuel flow setpoint until the airflow reaches the purge airflow rate.

4. Disengage the excess oxygen control loop and set the air flow rate setpoint at the purge airflow rate.

5. Continue to ramp the fuel flow down to a level corresponding to startup conditions and disengage the fuel flow loop. The burners should now be operating at the startup fuel flow setting. Place burner dampers in the correct startup position. Shut down each burner by closing its safety shutoff valve. Shutdown the last burner by tripping the main safety shutoff valve. Close all burner safety shutoff valves. Open all atmospheric vent valves.

6. Complete a unit purge.

7. Shut off the feedwater.

8. Shut down the blowers and optionally close the dampers.

Flow Charts of the Automatic Startup and Shutdown Procedures

Figures 15 to 24 show the flow charts of the startup and shutdown procedures.

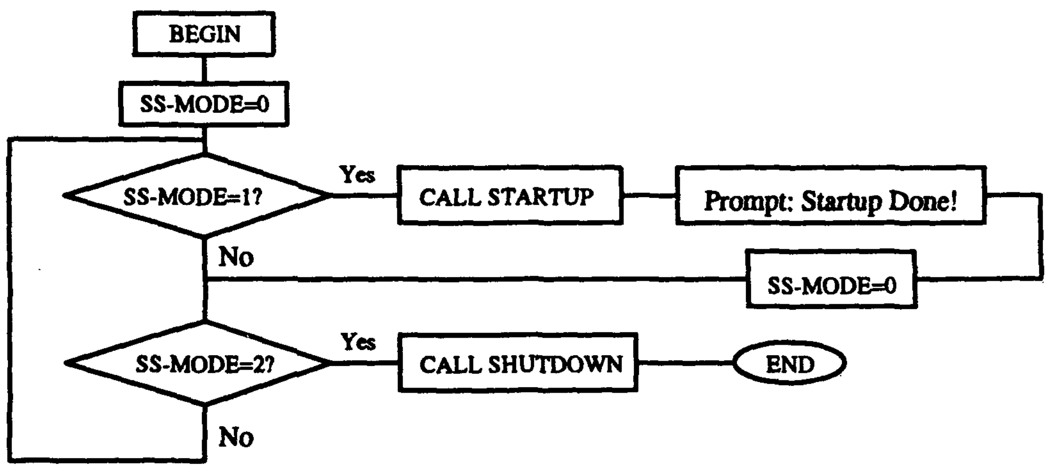


Figure 15. Flow Chart for Main Startup/Shutdown Procedure.

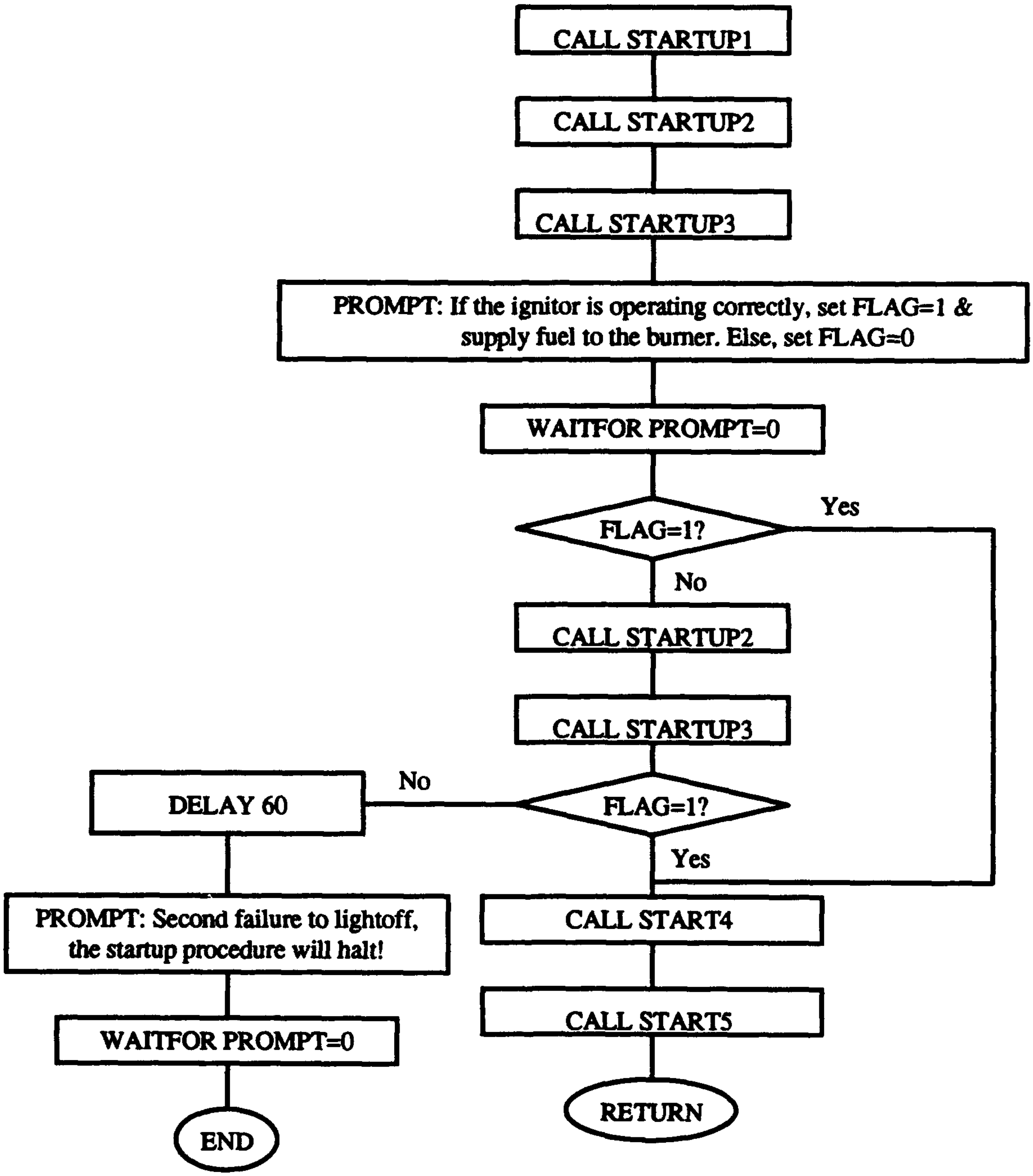


Figure 16. Flow Chart of Startup Subroutine.

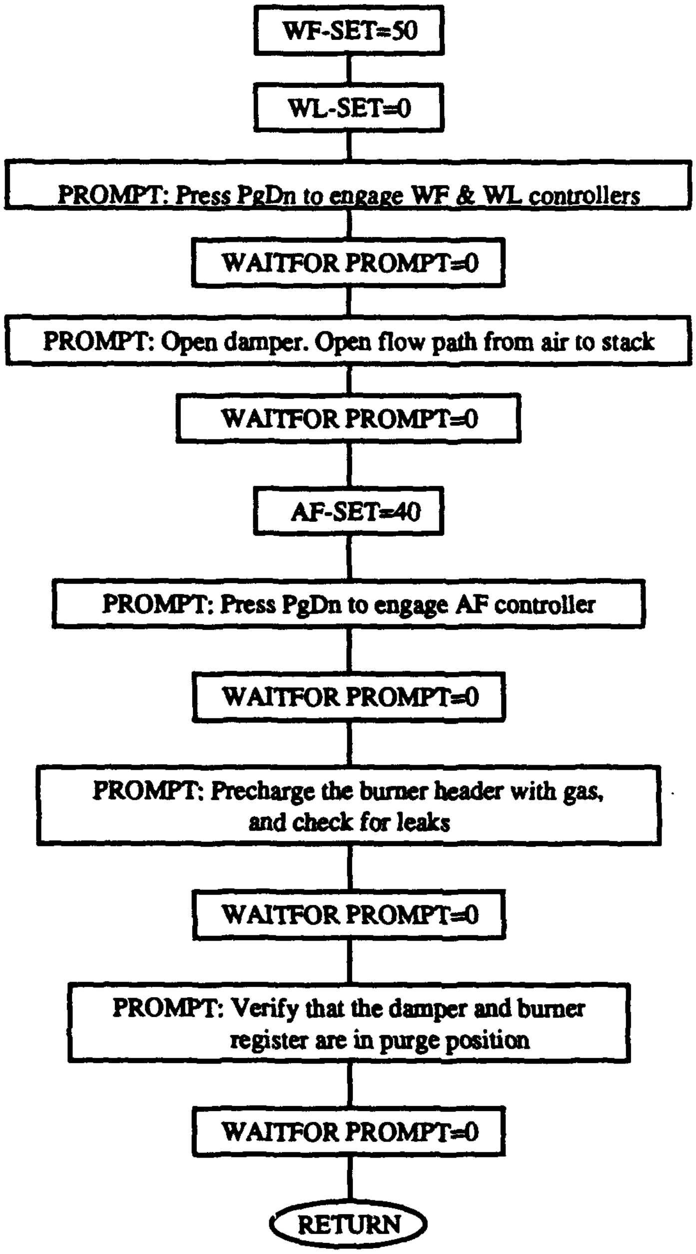


Figure 17. Flow Chart of Start1 Subroutine.

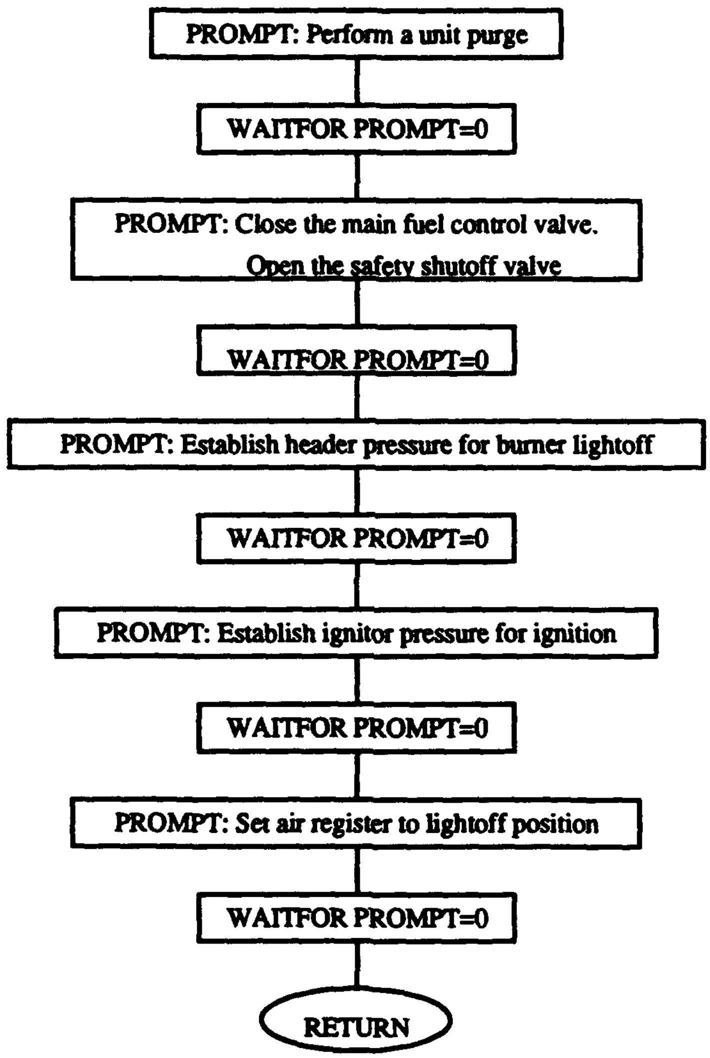


Figure 18. Flow Chart of Start2 Subroutine.

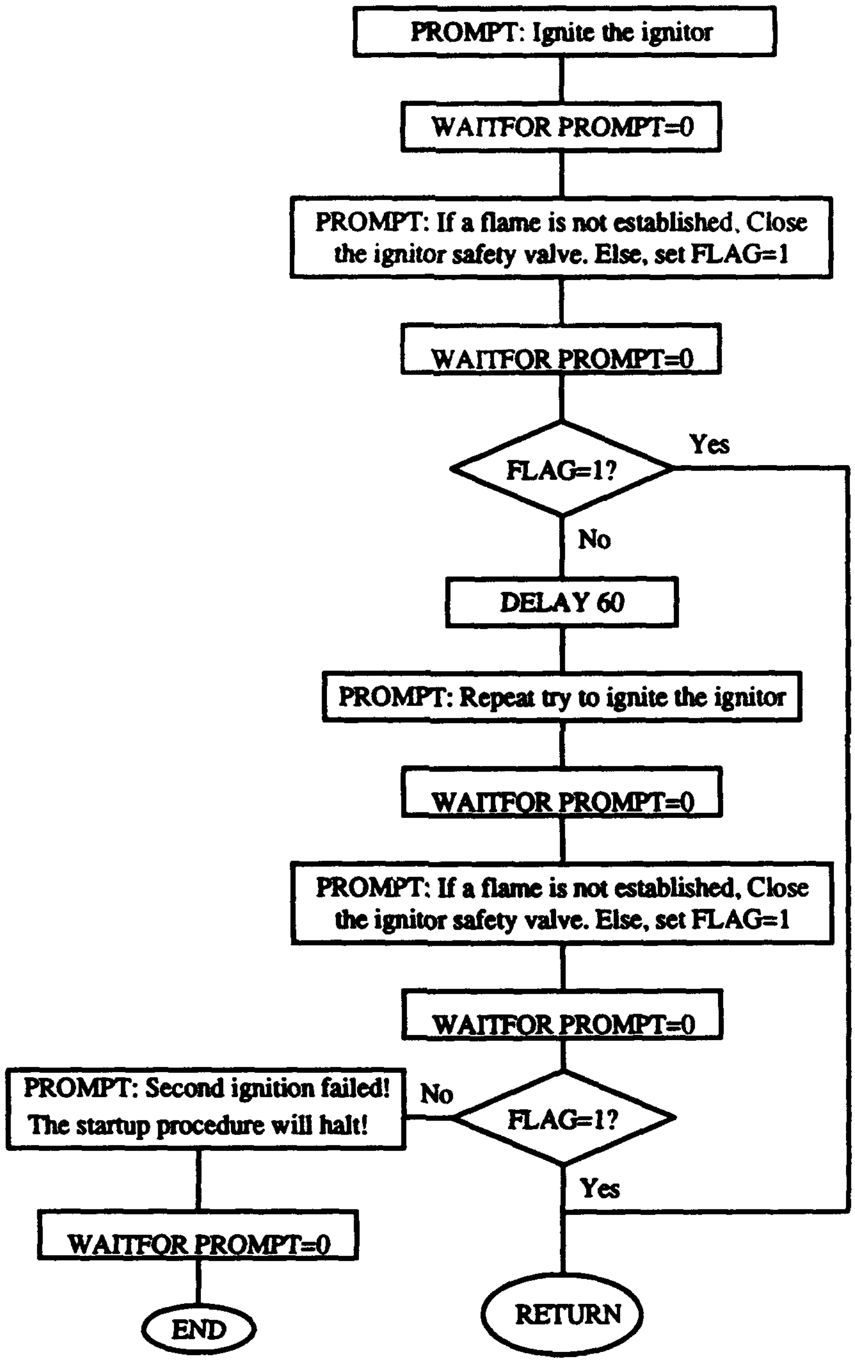


Figure 19. Flow Chart of Start3 Subroutine.

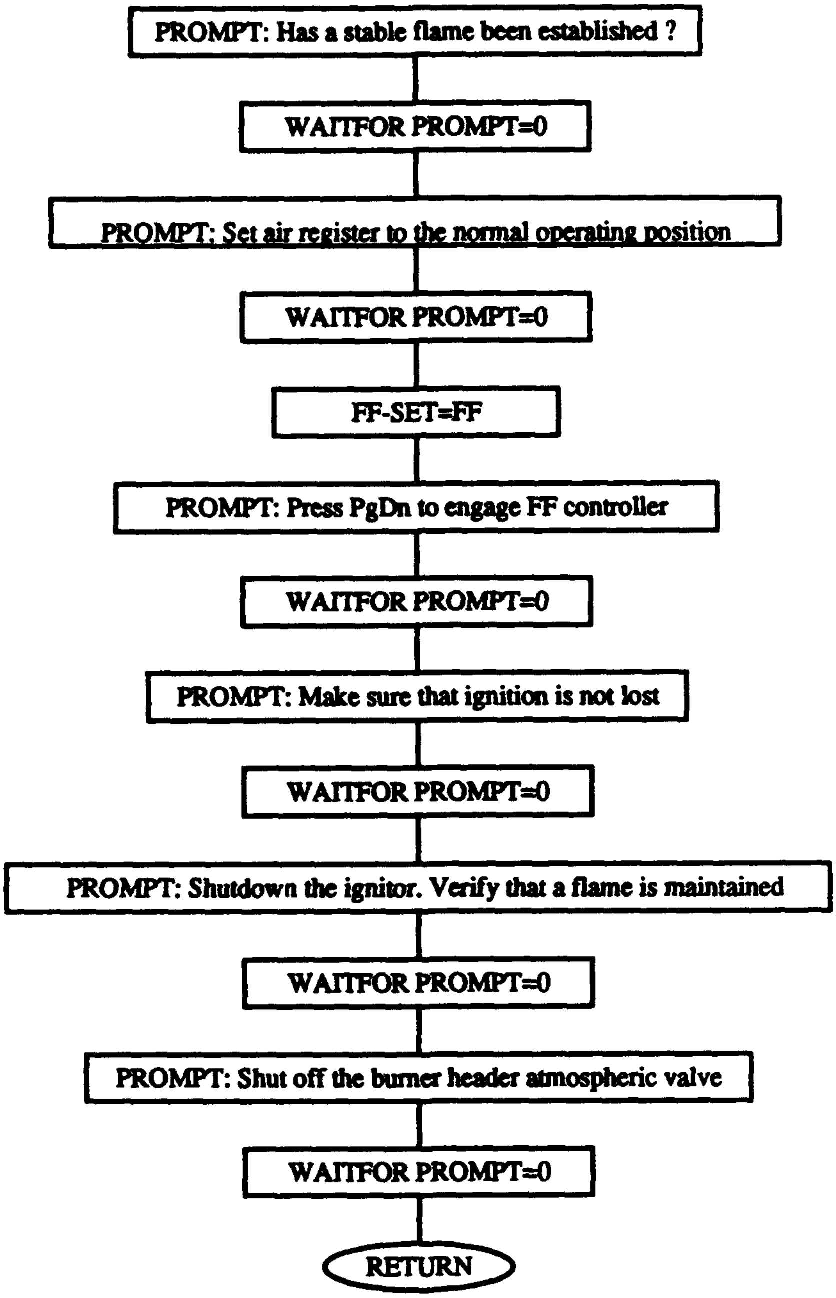


Figure 20. Flow Chart of Start4 Subroutine.

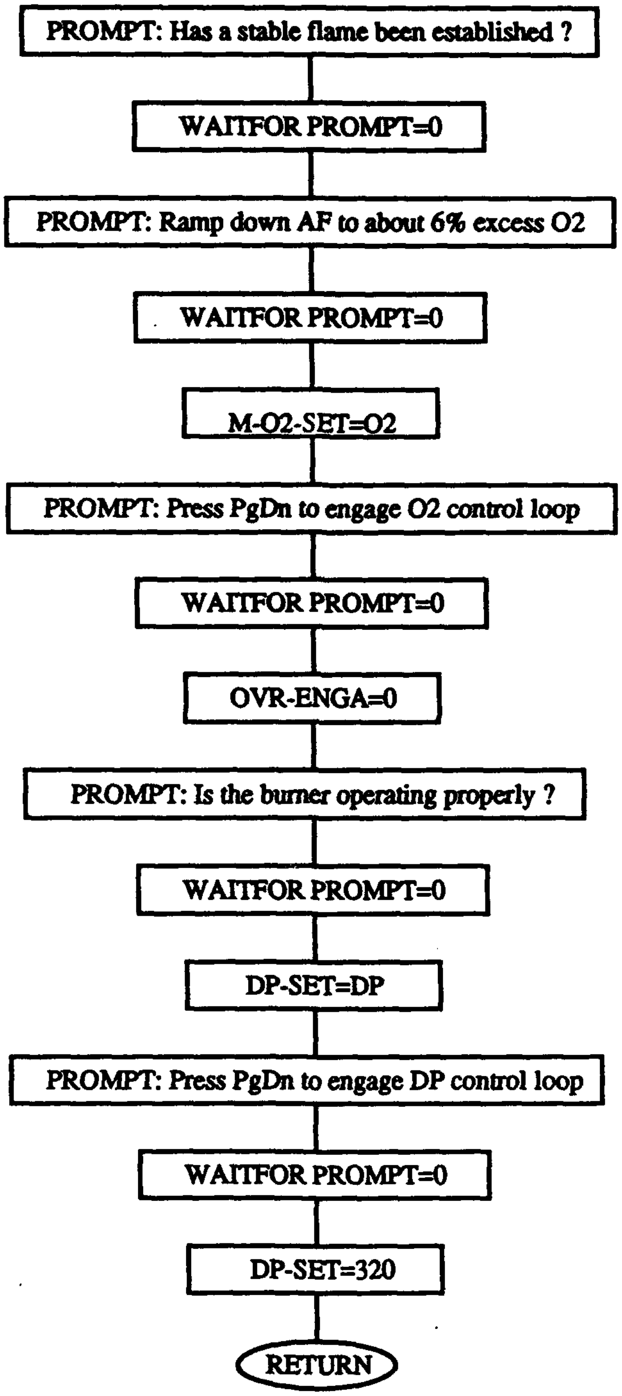


Figure 21. Flow Chart of Start5 Subroutine.

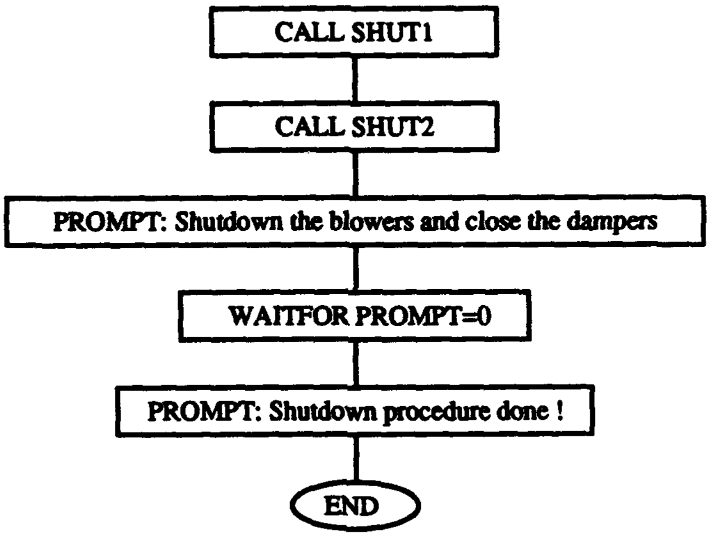


Figure 22. Flow Chart of Shutdown Subroutine.

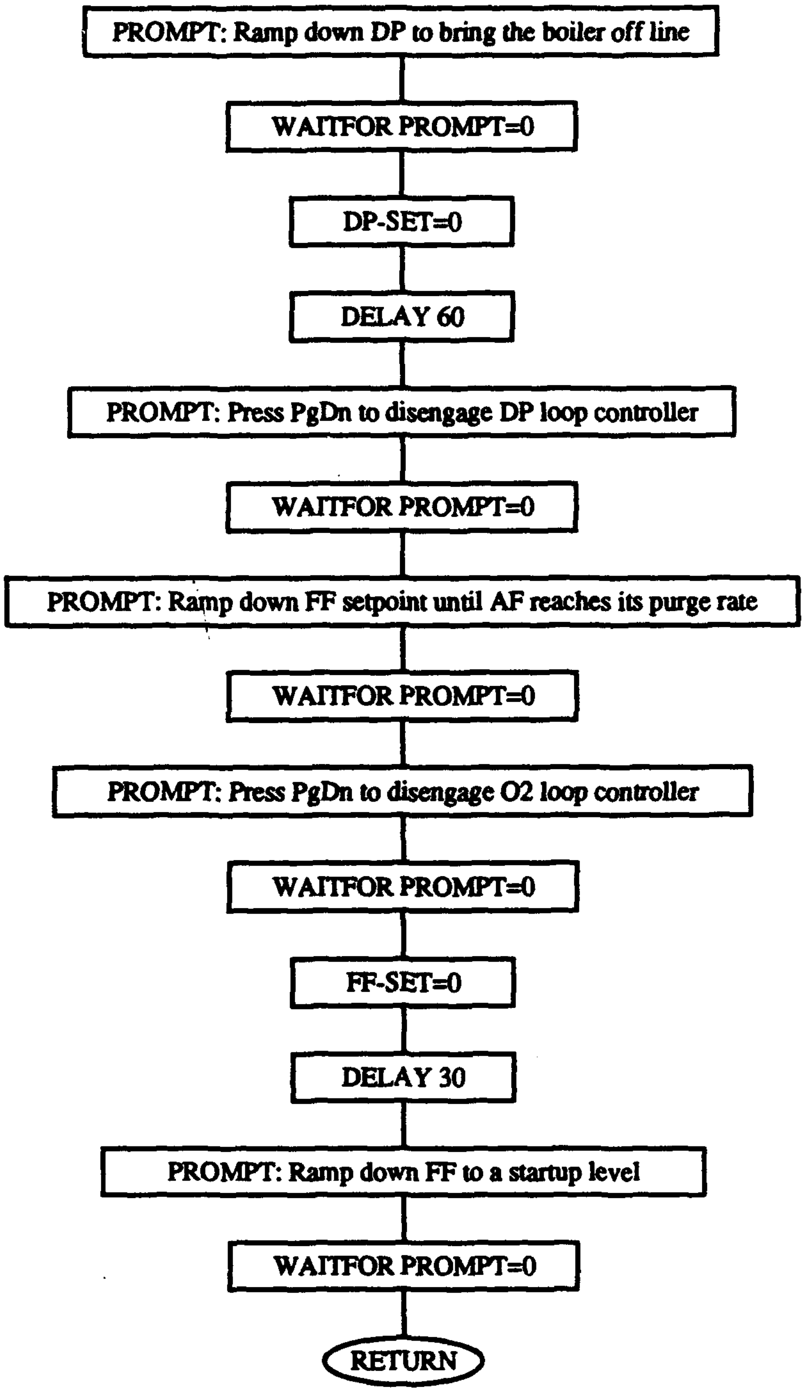


Figure 23. Flow Chart of Shut1 Subroutine.

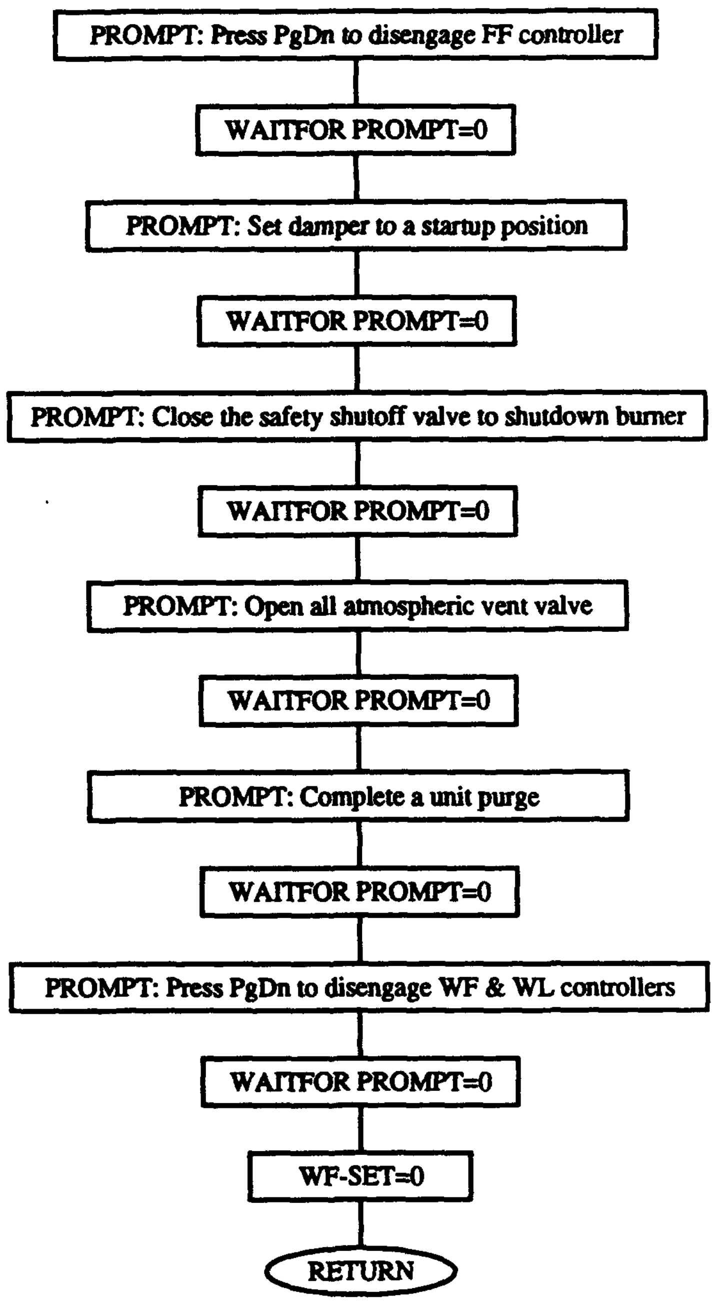


Figure 24. Flow Chart of Shut2 Subroutine.

Implement the Automatic Startup and Shutdown Procedures

FIX DMACS Graphic Environment. FIX DMACS provides a graphical environment that allows the user to monitor and control industrial processes using custom-designed graphical representations of the process. The graphical representation can be either a simple diagram or a detailed, animated display. Each display contains links that display messages, values, alarms, and real-time trends of the process. The FIX DMACS graphic environment developed for boiler automatic startup and shutdown consists of the following screens: BOILER STARTUP & SHUTDOWN, INNER LOOP PID, OUTER LOOP PID, and OUTPUT. All of these screens are linked together in the following order: BOILER STARTUP & SHUTDOWN --> INNER LOOP PID --> OUTER LOOP PID --> OUTPUT --> BOILER STARTUP & SHUTDOWN (Appendix D).

The BOILER STARTUP & SHUTDOWN screen is the master screen for boiler automatic startup and shutdown. In this screen, the options STARTUP or SHUTDOWN is given by a MAIN MODES field choice, and a sequence of prompts necessary for the startup or shutdown procedure are given (Appendix E). Moreover, all the setpoints of the process variables, WF (water flow), AF (air flow), FF (fuel flow), WL (drum water level), DP (drum pressure), and O_2 (stack excess oxygen), can be assigned on the screen. Furthermore, all the controllers can be engaged (or disengaged) by pressing the PgDn or PgUp key.

The INNER LOOP PID performs PID control for actuators, i.e., it controls the water flow (WF), air flow (AF), and fuel flow (FF) loops. In this screen, all the controller parameters can be easily set from the display. Also, all the setpoints of the actuators can be set on the display as well. The OUTER LOOP PID screen performs the same function as the INNER LOOP PID, except it controls the outer loop variables water level (WL), drum pressure (DP), and stack excess oxygen (O2). The OUTPUT screen graphically shows all the setpoints and measured values of the process variables in both numerical and bar graph format, and shows the historical trend of the setpoints and measured values of the drum pressure, stack excess oxygen, drum water level, and steam flow.

Operating Procedure. To perform the automatic startup and shutdown, the equipment required must include a µMac-6000 boiler controller as well as an IBM compatible PC. The PC should have boiler control software installed in a directory called BOIL, and the FIX DMACS package installed in the DMACS directory. The following steps outline the process:

a. Connect the COM1 port of the 386SX PC to the COM0 port of µMac-6000. Turn on the power of the µMac-6000.

b. Change to the BOIL subdirectory of the PC. This directory should contain the Kermit file transfer protocol and the executable boiler control code. Type KERMIT to establish the communications between the 386SX PC and the µMac-6000.

c. Type "SET BAUD 19200" to define the baud rate. Download the boiler control executable file to the µMac-6000 using Kermit.

d. Press CTRL-] (press "Ctrl" and "]" keys at the same time), then press "C" key to exit KERMIT. Type "EXIT" to return to the BOIL subdirectory.

e. Change to the DMACS subdirectory. Type "DMACS" to start the FIX DMACS program.

f. Choose the SAC4 function in the background menu. Type "Alt-x" to start the scan for node BOILER.

g. Choose the **PVIEW** function in foreground menu. Type "NEWBLR" to load FIX DMACS graphic environment. A screen named BOILER STARTUP & SHUTDOWN will appear on the display.

h. Move the cursor to the MAIN MODES field; type "1" for startup or "2" for shutdown. A prompt STARTUP NOW! or SHUTDOWN NOW! will appear on the screen.

i. Move the cursor to the field PROMPT RESPONSE, the program will display a sequence of prompts during execution of the automatic startup or shutdown procedure. Type "0" in the PROMPT RESPONSE field to move to the next step when the prompted task has been completed.

Furnace Draft Control Loops

The draft is defined as a "current of air." In balanced draft boiler systems, the furnace pressure is designed to be slightly negative relative to atmospheric pressure to ensure that any leakage will result in relatively cool combustion air leaking into the furnace instead of very hot combustion gases leaking out. Such boiler systems usually rely on the use of an induced draft fan in combination with a forced draft fan. The induced draft fan is used to reduce the furnace pressure and to ensure that it is always negative with respect to atmospheric pressure. In normal practice, the furnace pressure or draft is controlled to a very slightly negative pressure setpoint by regulating either or both the forced and induced draft fans. In the balanced draft boiler, the forced and induced draft fans share the load of moving the combustion air and flue gases through the system. The balance point is the pressure or draft in the furnace. This pressure level is determined by the relative amounts of "push" and "pull" of the forced and induced drafts, respectively. The furnace draft controller involves two control loops: the windbox pressure control loop and furnace pressure control loop. The object is to control the speed of the forced and induced fans such that the furnace pressure is maintained at a slightly negative setpoint. Conventional control methods for draft control mainly are PID control. Figure 25 shows the control logic diagram.

Since the system is subject to dynamic load change from the steam pressure demand, and the furnace draft measurement is subject to considerable process "noise," it is very difficult to tune the PID controller to get satisfactory results (Dukelow 1986). The proposed *Generalized Prediction Control* (GPC) schemes for the Furnace pressure control loop and Windbox pressure control loop (Figures 26 and 27)

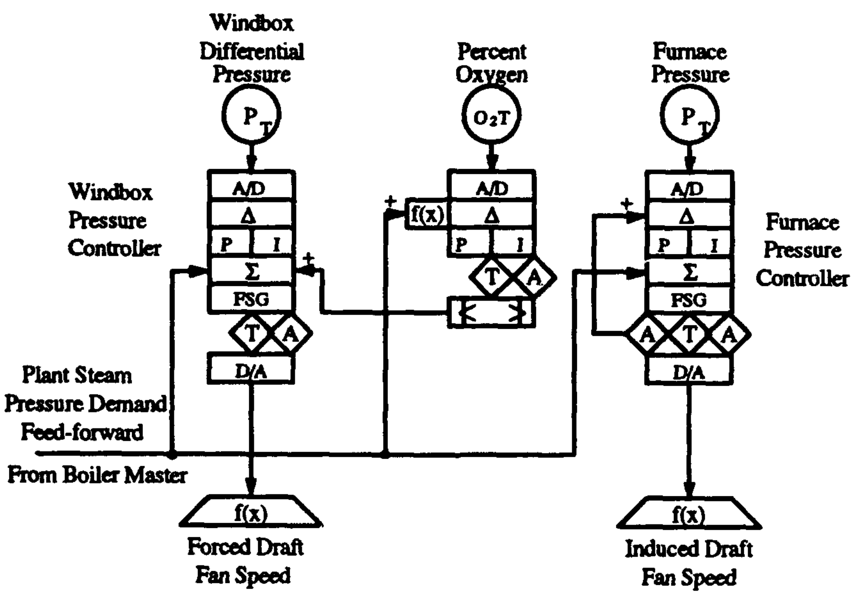


Figure 25. Draft Control Logic Diagram.

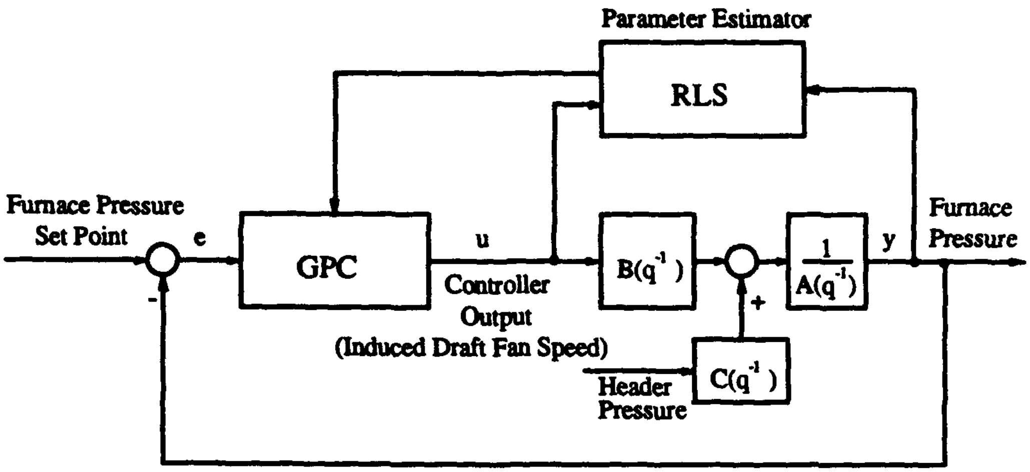


Figure 26. Furnace Pressure GPC Control Loop Diagram.

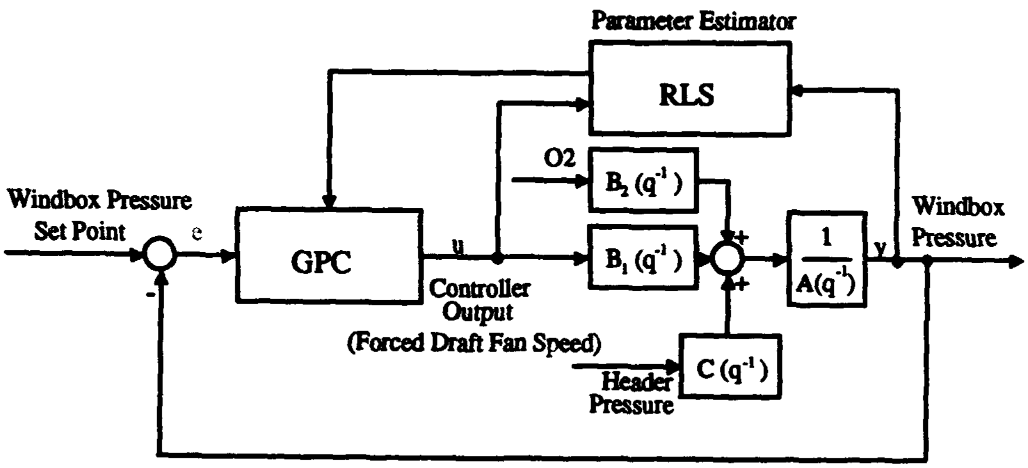


Figure 27. Windbox Pressure GPC Control Loop Diagram.

should improve system performance. Computer simulations used the proposed *Generalized Prediction Control* (GPC) algorithm to compare the performance by the PID control and the GPC control based on an artificial model of the processes. The model parameters were determined by off-line identification using the input-output data acquired from the plant (Åström and Wittenmark).

Furnace Pressure Control Loop

The furnace pressure model assumes that:

$$A(q^{-1})y(k) = q^{-n}B(q^{-1}u(k) + C(q^{-1})e(k) \qquad \text{[Eq 52]}$$

Where

y(k) is the furnace pressure
u(k) is the controller output, the induced draft fan speed command signal
e(k) is the disturbance from plant steam pressure demand.

Assume that the polynomials are:

$A(q^{-1}) = 1-.8959q^{-1}-0.043q^{-2}$
$B1(q^{-1}) = 0.2778 + 1.503q^{-1}$
$C(q^{-1}) = 1$

Let $\omega = 0.002$, and the disturbance signal e(k) from the steam demand be

$$e(k) = 0.1\left[\sin(\omega k) + \sin(3\omega k) + \sin(8\omega k) + \sin(15\omega k) + \sin(50\omega k)\right]$$

as shown in Figure 28.

Figure 29 shows the furnace pressure output using the GPC control scheme, where the desired furnace pressure is set as -0.5 in. of H_2O. It is clearly shown that despite the presence of the large disturbance (the signal-to-noise ratio is about 0.5), the output of the control system is close to the setpoint.

When a PID control scheme is used, the system performance is greatly influenced by the disturbance. If there is no disturbance, the PID control system can follow the setpoint very well (Figure 30). A disturbance however, will deteriorate system performance. Figure 31 shows the furnace pressure output with PID control, where the disturbance level is the same as with GPC control. The furnace pressure output oscillates with large tracking errors due to the large disturbance.

Windbox Pressure Control Loop

The simulation model for windbox pressure loop assumes that:

$$A(q^{-1})y(k) = q^{-n_1}B_1(q^{-1})u_1(k) + q^{-n_2}B_2(q^{-1})u_2(k) + C(q^{-1})e(k) \qquad \text{[Eq 53]}$$

Where
- y(k) is the windbox pressure
- $u_1(k)$ is the controller output, which is the forced draft fan speed command signal
- $u_2(k)$ is the feed forward signal of O_2
- e(k) is the disturbance from the plant steam pressure demand.

The polynomials are:

$A(q^{-1}) = 1-0.8959q^{-1}-.043q^{-2}$
$B_1(q^{-1})=.2778+1.503q^{-1}$
$B_2(q^{-1}) = 1$
$C(q^{-1}) = 1$

and loop delay n_1=1 second, n_2=0. Assume that the feedforward signal of O_2 is 6 percent, and the desired setpoint of windbox pressure is 1 in. of H_2O. Figure 28 shows the disturbance level from the steam pressure demand. Figure 32 shows the windbox pressure output when a PID controller was used. Because of the large disturbance (the ratio of the signal to noise is 0.5), the PID control cannot keep tracking the setpoint, so the system performance is unacceptable. Figure 33 shows the result with the GPC controller. Despite the presence of the large disturbance, the windbox pressure is maintained close to the setpoint 1 in. of H_2O.

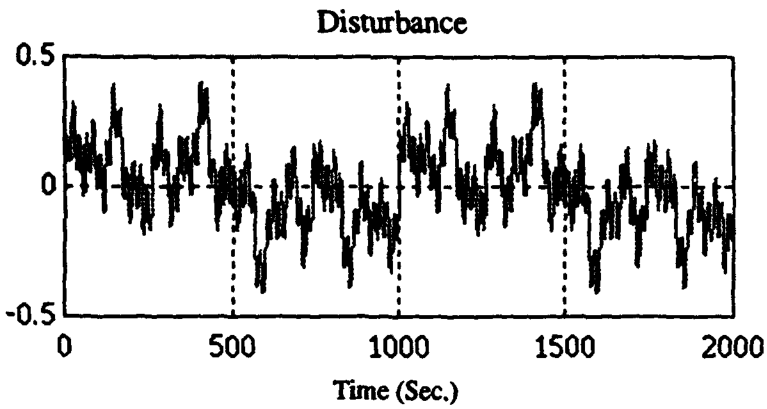


Figure 28. Disturbance e(k).

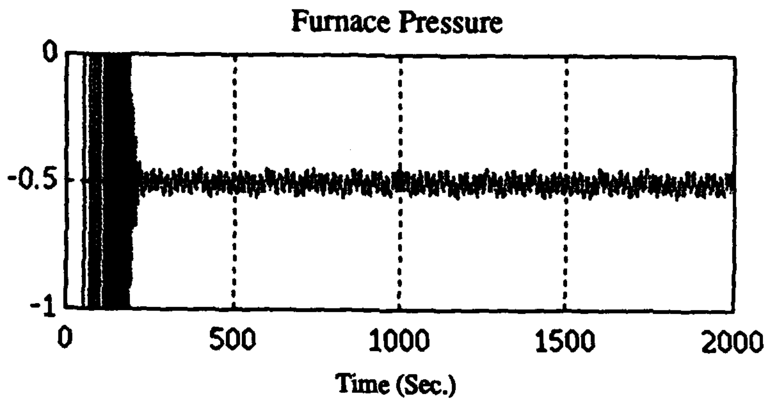


Figure 29. Furnace Pressure Output With GPC Control.

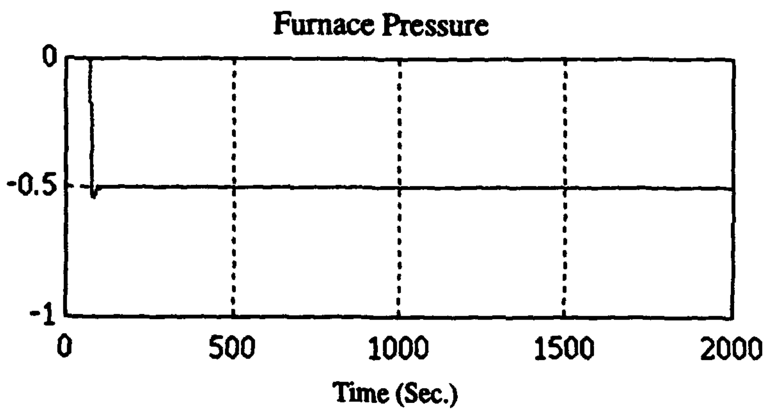


Figure 30. Furnace Pressure Output With PID Control, Without Disturbance.

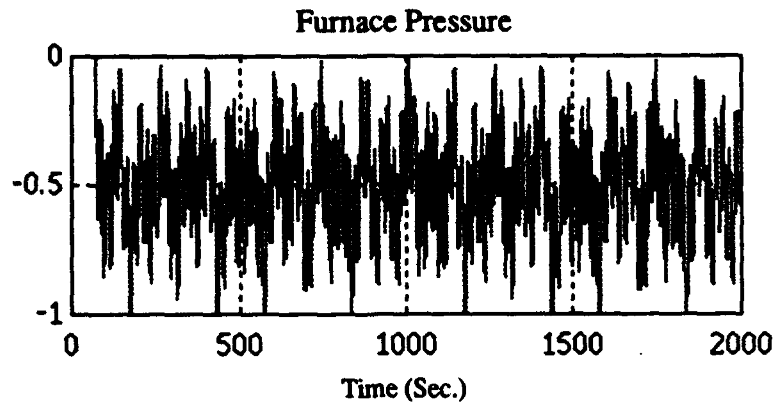


Figure 31. Furnace Pressure Output With PID Control.

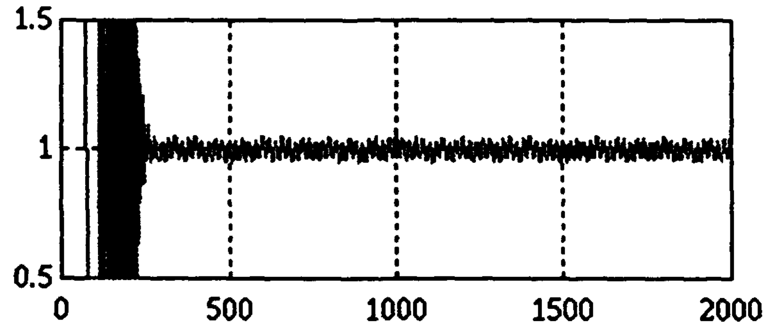


Figure 32. Windbox Pressure Output With PID Control.

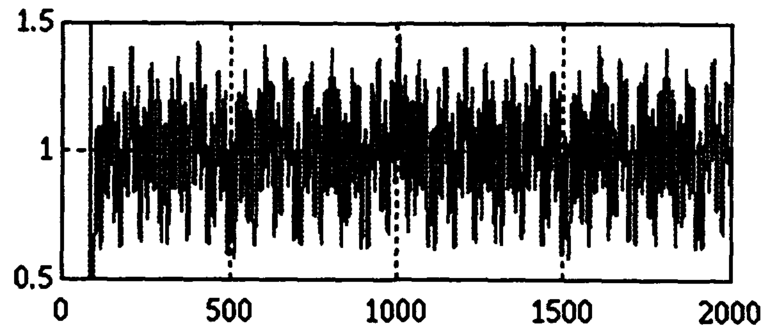


Figure 33. Windbox Pressure Output With GPC Control.

4 ABBOTT PLANT TEST RESULTS AND DISCUSSIONS

System testing was performed in the summer of 1991. The goal of the test program was to demonstrate that the prototype control system was able to successfully control an operational boiler. The test results would provide data for comparison of performance of the prototype controller with conventional control systems.

The subsequent tasks involve incorporating several enhancements to the control system to bring it closer to the level of performance needed for extensive field trials. These tasks include enhancements to the software to permit automatic startup and shutdown of the controlled boiler, and addition of "robustification" schemes (Bitmead, Gevers, and Wertz 1990; Middleman and Goodman 1990) to the model process identification algorithm used by the Generalized Predictive Controller (Clarke 1988; Clarke, Mohtadi, and Tuffs 1987). These robustification schemes are designed to ensure that the controller will operate well even under seldom encountered conditions. Finally, certain schemes designed to permit safe boiler operation in case of degraded boiler conditions as well as techniques for diagnosis of boiler operating equipment failure or degradation were to be identified.

Identification of System Deficiencies and Remedies

The current control system, based on an Analog Devices µMac 6000 (Analog Devices 1988, 1991) controller with an IBM PC type 286 or 386 host computer running the FIX software package, continues to serve as a successful development vehicle, even though several limitations have become apparent. Many of these limitations are a direct result of the fact that state-of-the-art commercial control equipment has not been designed to accommodate requirements of leading edge control schemes such as the Generalized Predictive Control algorithm. The major limitations are:

1. Communications capabilities between the PC running the FIX as the operator interface, and the µMac running the control code are limited. The amount of data that can be passed as one block of information is limited to 100 items. Of more concern is the fact that variables must be prescaled into a fixed range. This feature is designed to accommodate the needs of fixed range transducers and actuators, but it has been a major limitation when trying to transfer data such as the parameters in the plant model used by the GPC algorithm.

2. The moderate computing power of the µMac 6000 is a limitation. As reported elsewhere in this document, a higher sampling rate will be necessary to correctly control boilers of the capacity in use at Abbott Power Plant. In addition, several enhancements to the GPC controllers have been added. Both factors will place a heavier computing load on the µMac 6000. It is not clear yet if the µMac can meet these computing demands. It is clear that there is a limit that will most probably be reached at some point. It must be emphasized that as yet no computing limitations have been encountered. The µMac can easily control the boiler when configured as a conventional PID controller.

3. The µMac 6000 has two unfortunate functions that occur during a software upset or reset. The most important is the switching of all control outputs to minimum output level. Normally, outputs should be held at their last valid setting after a software upset. The second is the inability of the µMac to execute an orderly restart after an upset. Due to these limitations, the µMac cannot, by itself, be entrusted with boiler combustion control.

4. The µMac 6000 does not have a built-in power failure prevention system.

5. The memory used by the μMac 6000 lacks error detection and correction circuitry. This means that neither data nor code stored in random access memory (RAM) is protected against errors caused by bad memory locations, or bits lost due to cosmic radiation and power fluctuations.

Some limitations described above have already been addressed. Solutions to each limitation will be discussed below.

1. The limited communications ability of the FIX software package will be addressed by switching to Intellution's newer software package, the FIX DMACS (Intellution 1991). This package makes use of much better communication software and incorporates many other useful enhancements. The switch from the FIX to the FIX DMACS is straightforward. DMACS is delivered with tools to help convert applications from the FIX to the FIX DMACS.

2. The moderate computing power of the μMac 6000 is a continuing concern. The stated goal of this project has been to produce a set of control algorithms and methods that could be easily ported to other platforms. As a result, the computing limitations of the μMac 6000 are not seen as a major roadblock. If the computing capacity of the μMac 6000 is reached, another platform will be used. Therefore, a search for an alternate control device has begun. At present, the most promising candidate is GE Fanuc's Series 90-70 Model 781 PLC. This device is a high quality programmable logic controller with optional C programming capability. The processor on this device is an Intel 80386 running at twice the clock speed of the Intel 80188 on the μMac 6000. Clearly, this device should provide a large boost in computing power. In addition, many interlocks required by NFPA Standards 85A, 85B, and 85G can be implemented using the ladder logic capability of the 90-70. The 90-70 uses a special purpose custom processor to execute the ladder logic, so this function would have limited impact on the control algorithms. The series 90-70 is apparently the only currently available high quality PLC which includes a general purpose computing option adequate to meet the requirements of our system.

3. For testing purposes, Control Technology, Inc. Model 7312 Loopmate protection devices have been placed between the μMac controller and the boiler fuel, air, and water flow actuators. Upon a μMac watchdog timer time-out, the actuator command signals are frozen at the values they had 500 ms before the time-out. These devices also permit manual control of the boiler from their front panel.

The Loopmates probably could serve as a long-term solution to the problem caused by Mac outputs switching to low during an upset. A more serious problem is posed by NFPA Standard 85B, "Prevention of Furnace Explosions in Natural Gas-Fired Multiple Burner Boiler-Furnaces 1989 Edition," section 4-2.2.3, which states that, "The logic system performing the safety functions for burner management shall not be combined with any other logic system." This means that a second control device must be dedicated to burner management. This standard may not apply to some central heating plants. NFPA 85A, "Prevention of Furnace Explosions in Fuel Oil- and Natural Gas-Fired Single Burner Boiler-Furnaces, 1987 Edition" may apply to many central heating plants. NFPA 85A does not require an independent burner management controller. The requirement of NFPA 85B is prudent, and because a standardized control system is to be designed, it would appear wise to satisfy the requirements of NFPA 85B.

As a result, it may be best to insert a programmable logic controller (PLC) between the μMac and the boiler. The PLC will monitor and pass through the fuel and air command signals. As required by section 4-2.2.3 of NFPA 85B, the PLC will be programmed to implement purge interlocks and timing, mandatory safety shutdowns, trial timing for ignition, and flame monitoring. In addition, the device will be programmed to test for dangerous air/fuel ratios and to lock the air and fuel command signals upon a μMac watchdog timer upset. The PLC will itself be chosen to meet the requirements of NFPA 85B, including the requirements of sections 4-2.2.1 and 4.2.2.2. At present, it would appear that a simple PLC can meet the requirements of NFPA 85B. One good candidate is the GE Fanuc Series 90-30 Model 311 PLC.

GE Fanuc has further expanded its Series 90-30 PLC CPU family with the addition of the mid-range Model 341. The new 341 CPU includes an 80188-based processor with special coprocessor that operates at an accelerated clock rate of 20 MHz compared to the previous 8 MHz. The 90-30's memory has been enlarged by a factor of five, to 40K words. GE Fanuc also has increased the number of registers from 2,000 to 10,000. The increased clock speed boosts the Boolean I/O scan rate from 0.4 msec/K to 0.3 msec/K. And non-Boolean operations occur in half the time previously required. Analog inputs for the 90-30 have increased from 128 to 1024, while its analog outputs have climbed from 64 to 256. This increased speed and I/O capability means that the 90-30 can be used in high-speed distributed applications such as high speed/large quantity material handling, which previously required a more expensive and larger PLC.

GE Fanuc's high-end, open-architecture, 90-70 PLC also improved its CPU performance. One big advantage to using open systems and industry standards is the relative ease of upgrading system processors and support peripherals while maintaining a seamless upgrade path.

The 90-70 now includes a 80486DX-based Model 914 CPU. As a result, this CPU has the same Boolean scan rate (0.4 msec/K) as other 90-70 CPUs, but now includes the capability to handle higher level math functions more quickly, and features an expanded user memory of 512K bytes (GE Fanuc Automation 1993). To help ease system maintenance, the Model 914's operating system can be downloaded from disk to flash PROM, to ease and simplify O/S updates. The Model 914 is intended for more complex control applications and can be programmed in relay ladder logic, state logic, sequential function block (SFC), and in C—like the other 90-70 CUPs—to suit individual program development needs.

The Model 914 CPU is capable of industry-standard communications protocols, including Ethernet, MAP, RTU/Modbus, CCM, SNP and GE Fanuc's Genius intelligent, distributed I/O system, and the I/O link. The last provides an interface between GE Fanuc PLCs and its computer numerical controls. Even if the μMac is replaced with a programmable logic controller with computational capability (see above), a second burner management controller will be required.

4. The fact that the μMac 6000 (and the Loopmate devices currently used to ensure that the actuator command signals are held at reasonable values after an upset) does not have inherent power failure protection, should not be a cause for much concern. The simple addition of an uninterruptable power supply (for example, a Superior Electric Company Model UPS61005R Stabiline) can easily ensure control system operation for the required time.

5. The lack of memory error detection and correction circuitry on the μMac controller is of some concern. If the μMac is replaced with a more powerful computing device, one of the selection criteria for the replacement device will be memory protection. The problem is not as severe as it might first appear. In the first place, the addition of a line conditioner and uninterruptable power supply such as the unit discussed above will greatly lessen the likelihood of memory corruption due to power fluctuations. While the RAM memory of the μMac has neither error correction nor error detection circuitry, it is static and not dynamic RAM. Static RAM is far more resistant to corruption due to power fluctuation and high energy particles than is dynamic RAM. Finally, the addition of a burner management controller as required by NFPA 85B along with the μMac's watchdog timer capability provides a measure of redundancy and protection against failure of the μMac controller due to memory corruption.

Testing of the Prototype Control System

Outline of the Test Plan

The following is an outline of the test sequence at the Abbott Power Plant boiler no. 2:

1. Stabilize the boiler at an acceptable operating point. The excess oxygen level would be set high as a safety measure.

2. The fuel, air, and water flow actuator loops were tuned next. These loops are conventional PID control loops implemented digitally on the Mac controller.

3. Once the actuator control loops were tuned, each outer loop control model was identified, and after identification, the controller was started. The first outer loop model to be identified was the water level loop. After convergence of this model, the GPC controller for water level was started. Next, this procedure was repeated for excess oxygen. Finally, the drum pressure control model and controller was started.

4. Next small perturbations of setpoint for each control loop were applied to test transient response.

5. A steam demand disturbance was simulated by changing the fuel rate on boiler no. 3 (on the same steam header).

6. A large change in steam demand was initiated next by making a large, gradual change in the fuel rate of boiler no. 3. As a safety measure, the GPC controller was inactive during the transition. The boiler was in manual control mode during the transition. After reaching the new operating point, steps 2 through 5 were repeated. The principle goal of this procedure was to determine the changes in the controller model at different operating points.

7. The boiler was returned to the original set of operating conditions with the controllers on. This experiment demonstrated the self-tuning tracking ability of the controllers.

8. The perturbation experiment was repeated at this operating point.

9. Finally, this entire test plan was repeated using digital PID controllers to create a comparison data set.

Appendix A gives details of this test plan.

Test Difficulties

Problems Tuning Actuator PID Loops. The parameters of the PID controller were tuned by using standard tuning rules from Åström (1984) (Appendix B). Using the transient-response method, the controllers worked as expected except that the controlled variables contained limited cycle oscillations. The pneumatic PID controllers normally used in Abbott also produce such oscillations in the controlled variable, but at a smaller amplitude. Much time was spent in an effort to satisfy the operators that actuator control was acceptable. The solution to the larger than normal limit cycles is probably a faster sampling rate.

Problems With the Identifier on the Water Level Loop. It was found to be impossible to obtain convergence for plant model-parameters associated with the water level loop. Since the GPC is an adaptive model-based method, the efforts to use the GPC on the water level loop were unsuccessful. Instead, a fixed plant model with predictive control for the water level loop was used for the test.

Test Results

Drum Pressure Loop. The drum pressure loop performed well under GPC control (Figure 34). Though the setpoints were varied over a wide range (from 323 to 347 psi in less than 1000 seconds), the loop output tracked the setpoint nicely. This is quite difficult for conventional pneumatic PID control. Tighter drum pressure control can improve upset recovery and operational safety. The Abbott Power Plant personnel all agreed that the drum pressure control was superior using the GPC algorithm.

Excess Oxygen Loop. In general, the performance of the excess oxygen loop under GPC control is excellent (Figure 35, data batch 2000-6000 seconds). During the sharp reduction in boiler steam flow that occurred after 6000 seconds, control degraded considerably. It was found that this problem mainly was caused by a deficiency of the identifier. The excess oxygen loop is nonlinear so the model dynamics change at different operating regimes. To check this, one might choose a batch of data from 5001 to 7500 seconds, and use the PEM batch parameter estimation function in MATLAB to estimate the model parameters. Under these conditions, the maximum one step ahead prediction error is about 2.8 percent of oxygen (Figure 36). By shifting the data set from 6001 to 8500 seconds, however, the maximum one-step-ahead prediction error would be only about 1.6 percent of oxygen (Figure 37). Clearly, the model parameters needed to change sharply after 6000 seconds into the test. Since a slow sampling rate (5 seconds for excess oxygen loop) was used during the test, the identifier could not quickly catch the rapid changes of the parameters in the model so that the prediction control algorithm was no longer able to control the excess oxygen level satisfactorily. Using the data set logged during the test, it has been discovered that the identifier will track the parameters well when the process is sampled at a 1-second update rate.

Water Level Loop. Test results for this loop are marginal (Figure 38), since a fixed model and a predictive controller were used. Nevertheless, the performance of the GPC controller is at least as good as the continuous pneumatic PID controller in the Abbott power plant. Figure 39 shows the results using PID controller. After the test, analysis shows that parameter estimates will converge after 40 minutes at a sampling rate at 20 seconds (Figure 40). At a faster sampling rate, the parameter estimates converge much quicker. Figure 41 shows that, at a 1-second sampling rate, the estimated parameters converge after 12 minutes.

Oscillatory Behavior of Actuators. The fuel, air, and water flow actuator loops demonstrate oscillatory behavior (Figure 42). The main cause for this was probably the backlash in the drive mechanism. The oscillation was also present when using the Abbott Power Plant continuous pneumatic PID controllers, but at a smaller amplitude. A fast update rate would reduce the amplitude of this oscillation, but it is likely that only improved actuation equipment could eliminate such a hard nonlinearity.

Change in Time Delay With Operating Point. Time delay is an important factor in choice of model structure. One effective approach to determining the process time delay is to observe the step response of the process. In the case of the excess oxygen loop, it seems that the process delay is slightly changed at different operating points. The change was surprisingly minor (only 1 to 2 seconds); a large time delay change had been anticipated. It appears that gas buoyancy dominates the gas flow rate in the boiler. For the data batch 6501-7000 seconds, the correct time delay was found to be 13 seconds for fuel flow setpoint to oxygen, and 8 seconds for air flow setpoint to oxygen (Figure 43). If the time delay is increased to 14 seconds for fuel flow setpoint to oxygen, Figure 44 shows that the shape of the step response becomes unreasonable. Further, for data batch 7001-7500 seconds, the correct time delay is

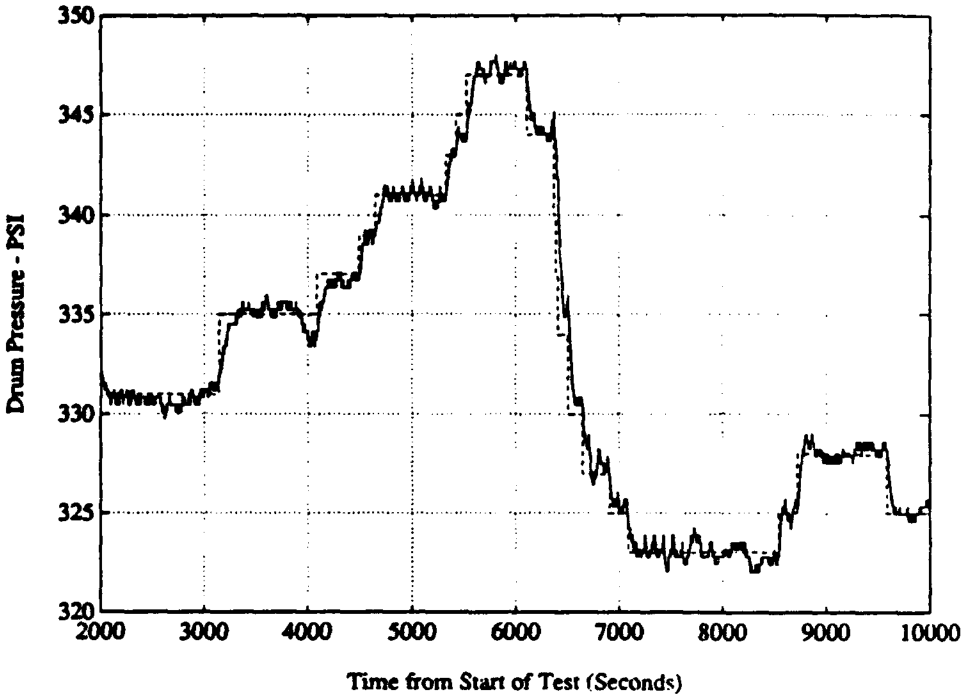


Figure 34. Drum Pressure and Set Point With the GPC Control.

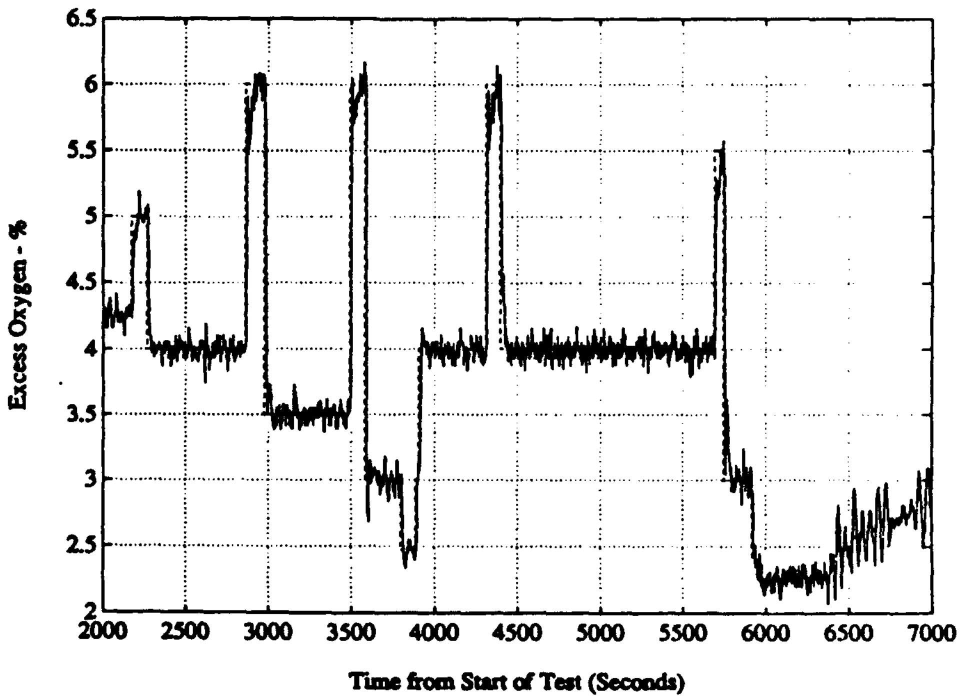


Figure 35. Exhaust Oxygen and Setpoint With GPC Control.

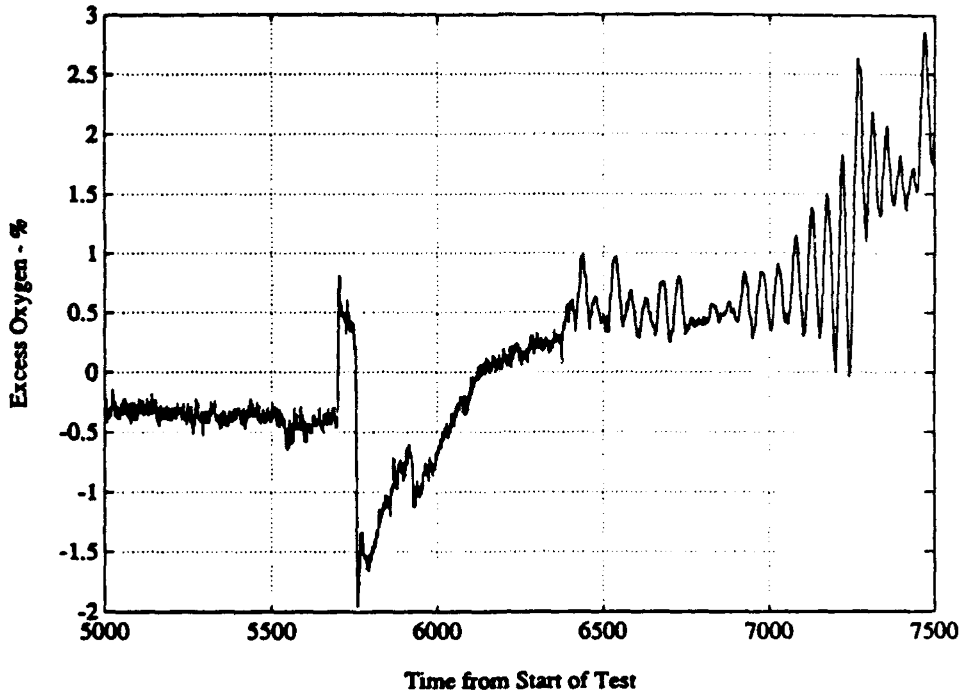


Figure 36. One Step Prediction Error for the Data Batch 5001 to 7500 Seconds.

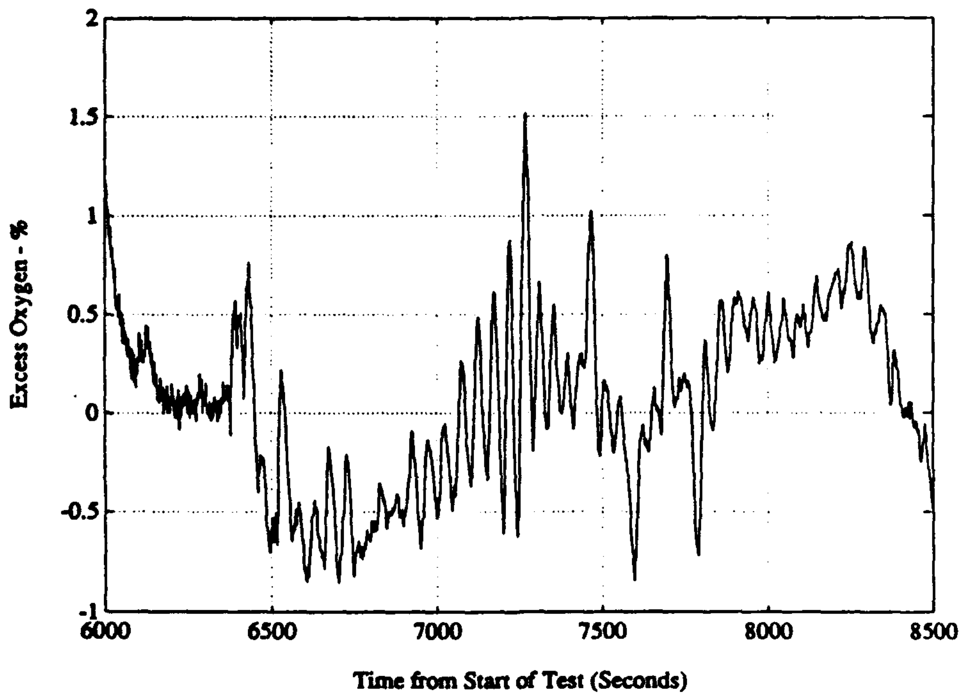


Figure 37. One Step Prediction Error for the Data Batch 6001 to 8500 Seconds.

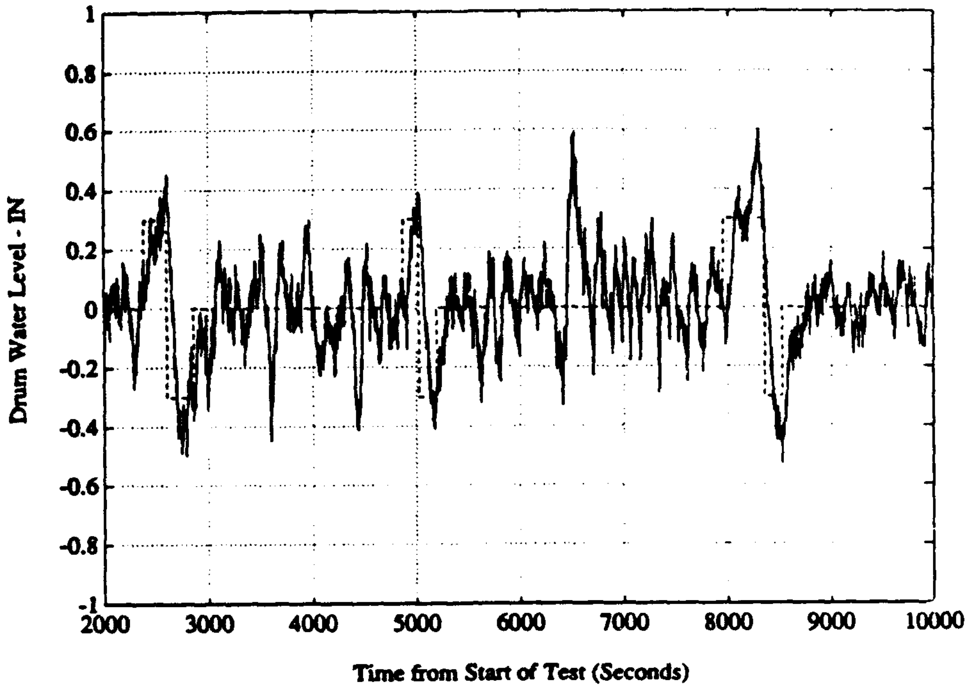


Figure 38. Drum Water Level and Setpoint With GPC Control.

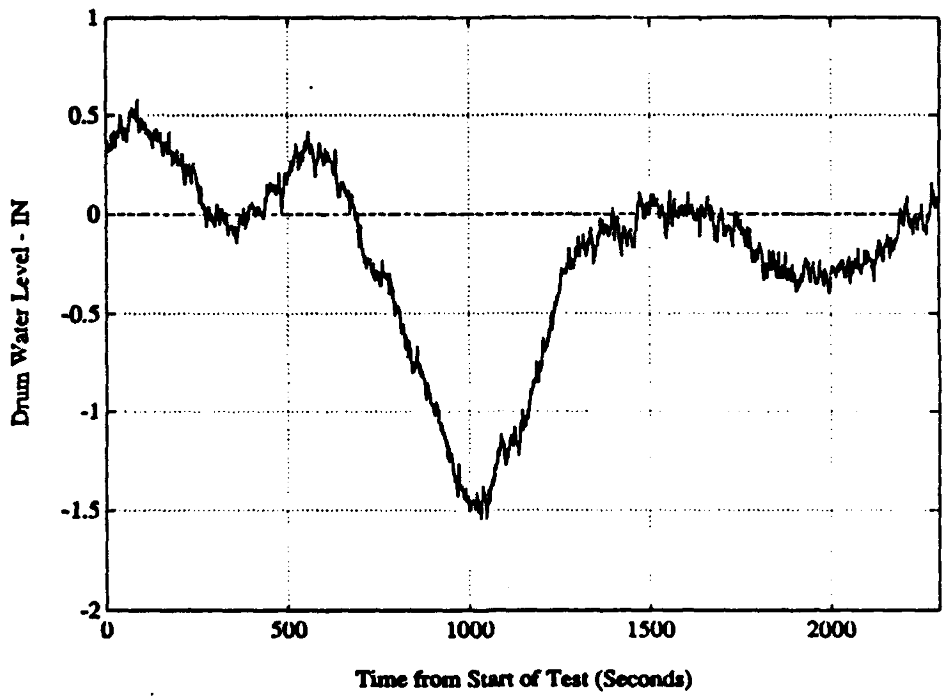


Figure 39. Drum Water Level and Setpoint With PID Control.

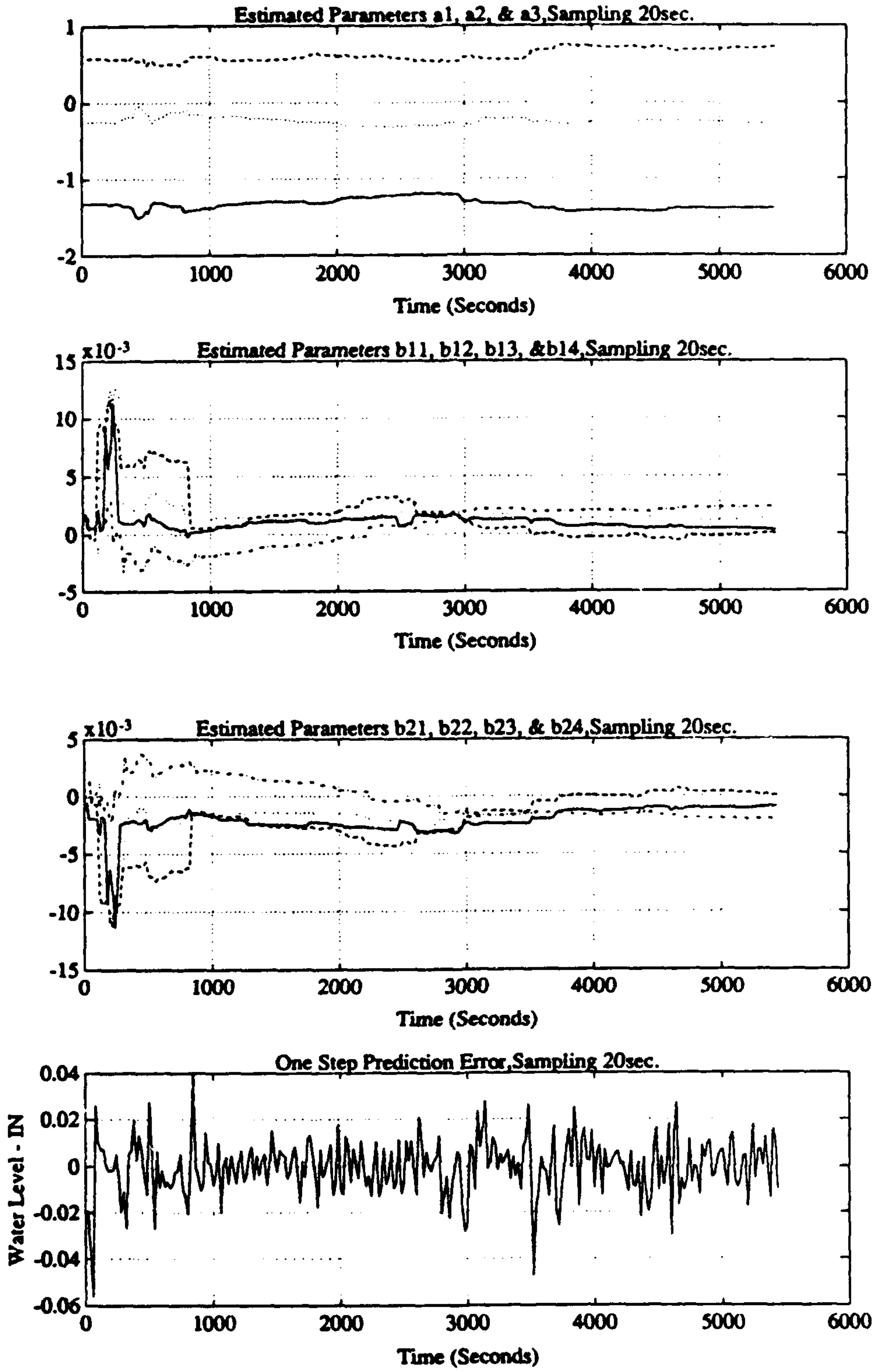


Figure 40. Estimated Parameters and Prediction Error With 20-Second Sampling Period.

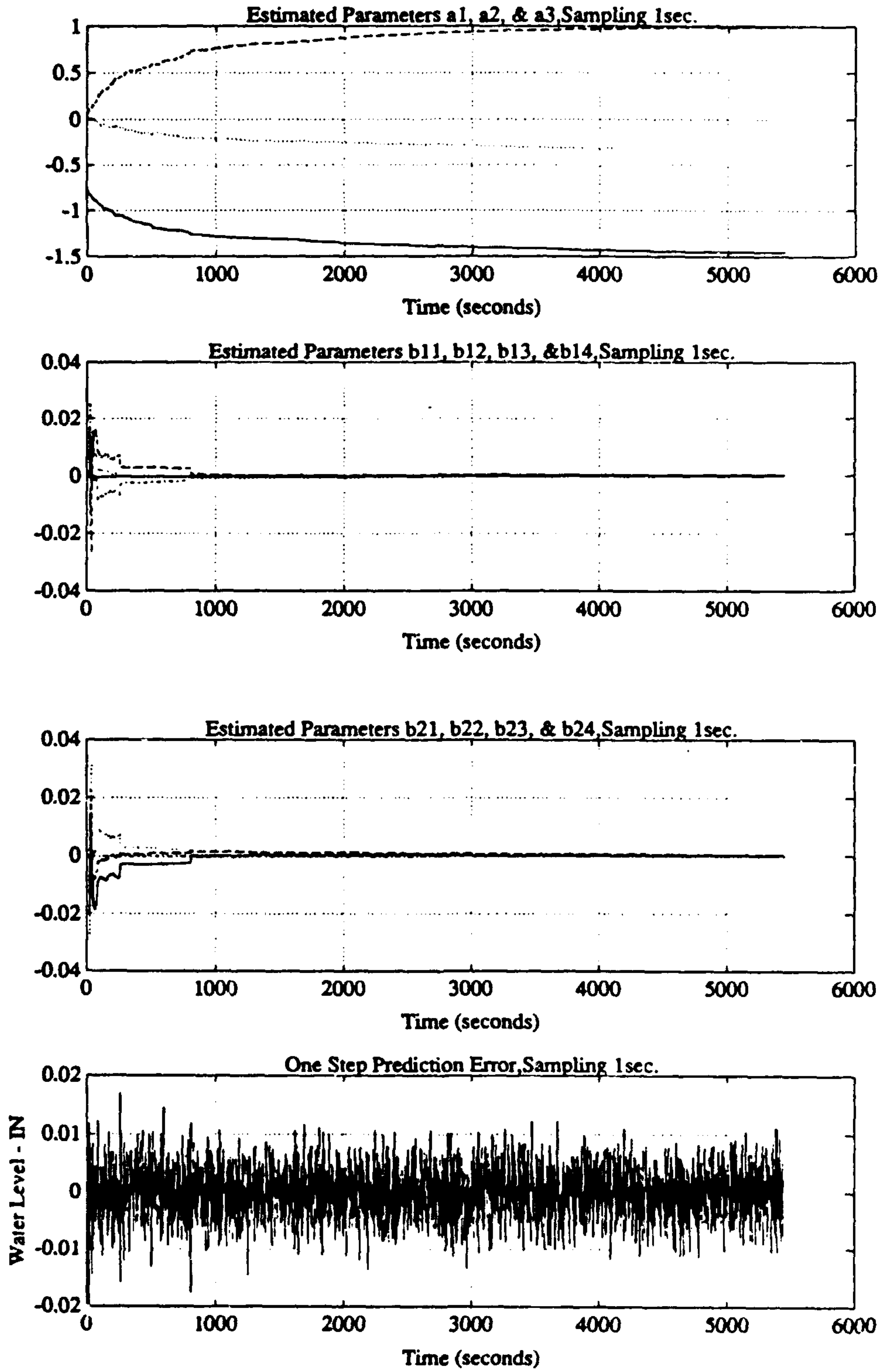


Figure 41. Estimated Parameters and Prediction Error With 1-Second Sampling Period.

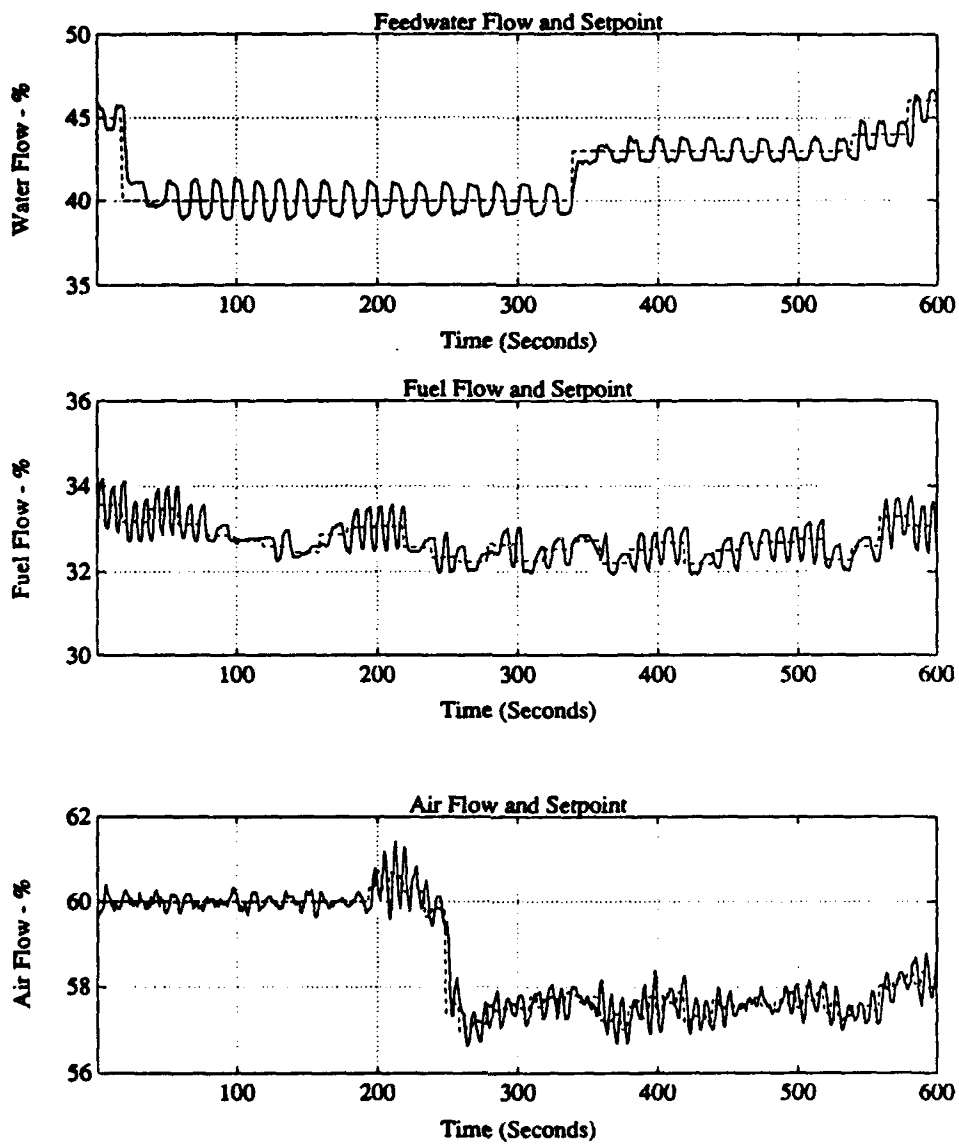


Figure 42. Oscillatory Behavior of the Actuators.

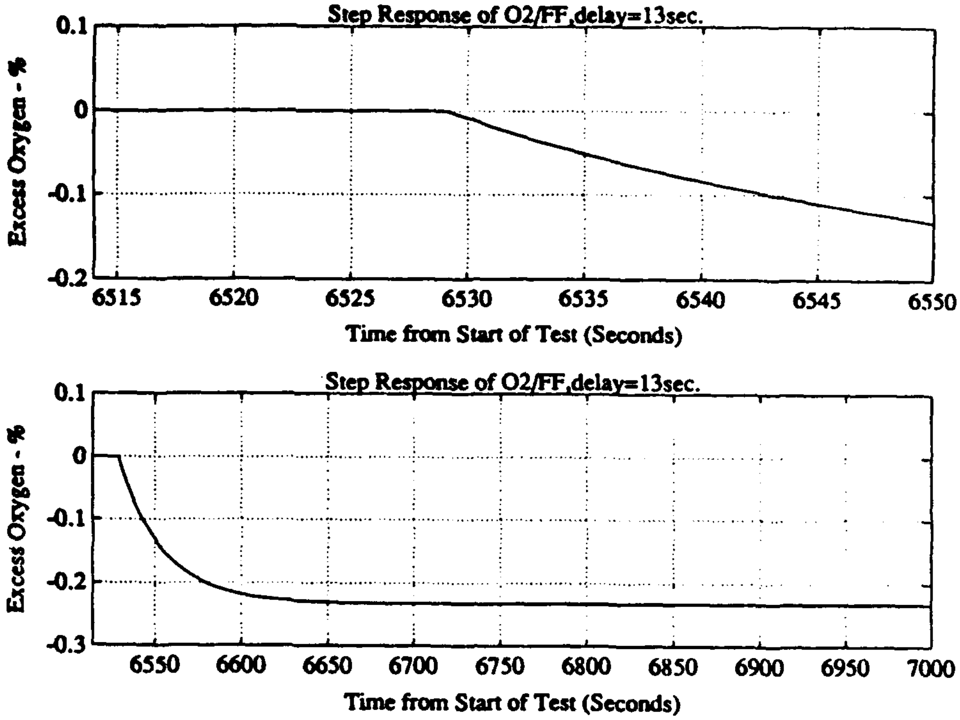


Figure 43. Step Response of Data Batch 6501 to 7000 Seconds of O2/FF; Delay=13 Seconds.

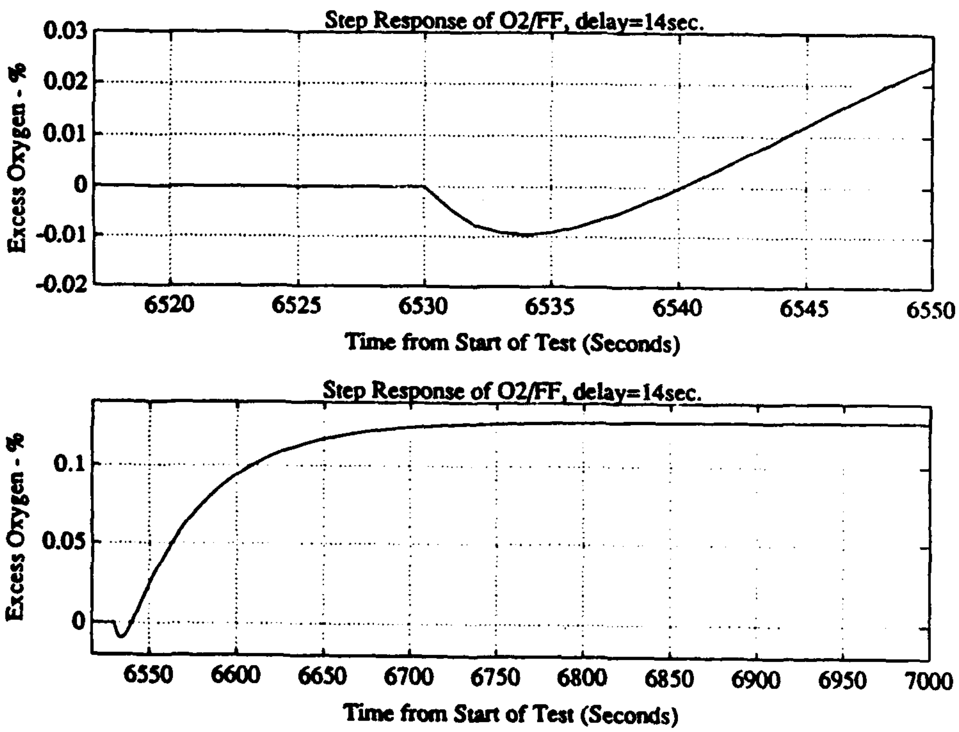


Figure 44. Step Response of Data Batch 7001 to 7500 Seconds of O2/FF; Delay=14 Seconds.

changed to 14 seconds for fuel flow setpoint to oxygen (Figure 45). If the chosen time delay is 15 seconds, the step response still looks reasonable (Figure 46). If the time delay is increased further, Figure 47 shows that the step response of the process diverges.

Performance Comparison of GPC and PID. Directly comparing the performance of the GPC controller with PID controllers proved to be difficult. Mainly this is due to the plant not being instrumented for setpoint measurement when the conventional pneumatic PID controllers were in use. It was impossible to match the operating conditions, but a limited comparison could still be made. Figure 48 shows the exhaust oxygen and setpoints with the pneumatic PID controller and the GPC controller. It is clear that with the GPC control, the step response of the exhaust oxygen (within 0.2 percent O_2) is much tighter than with PID control (within 0.5 percent O_2). The GPC algorithm provides excellent control of combustion. This, potentially, has large payback. Tight oxygen control reduces losses due to excess hot gases leaving the stack to a very low level. Exhaust gas CO control can carry this benefit a step further. The system under development will be cost competitive with simple oxygen trim equipment, but will permit the tightest possible combustion control with the resulting economic benefits.

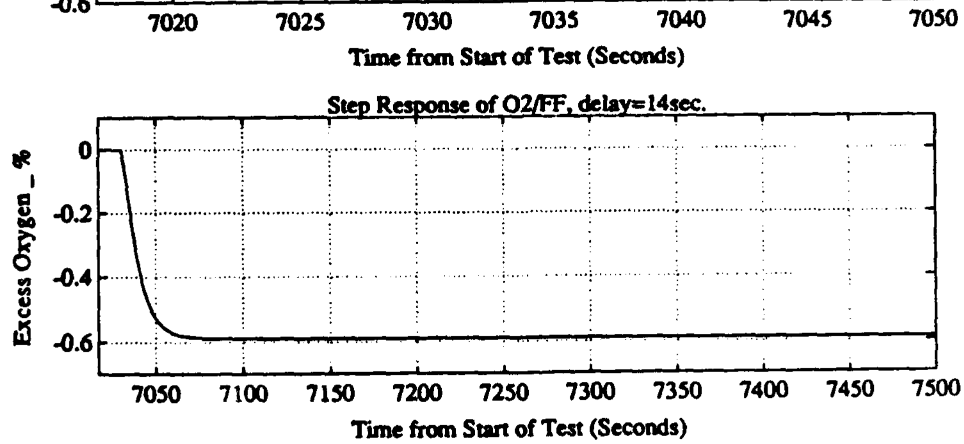


Figure 45. Step Response of Data Batch 7001 to 7500 Seconds of O2/FF; Delay=14 Seconds.

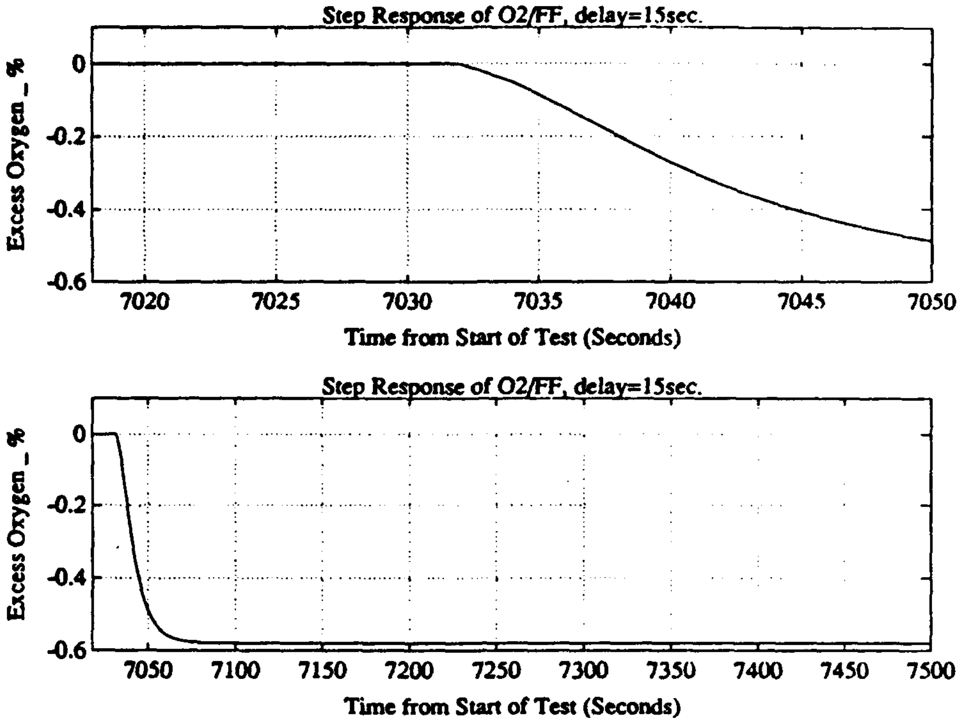


Figure 46. Step Response of Data Batch 7001 to 7500 Seconds of O2/FF; Delay=15 Seconds.

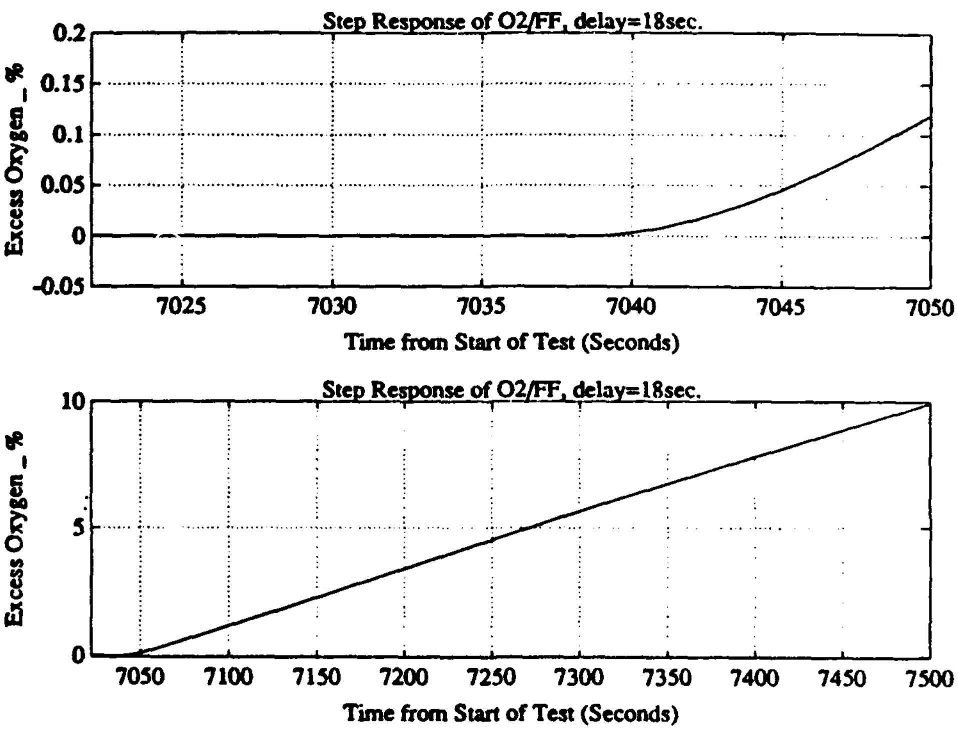
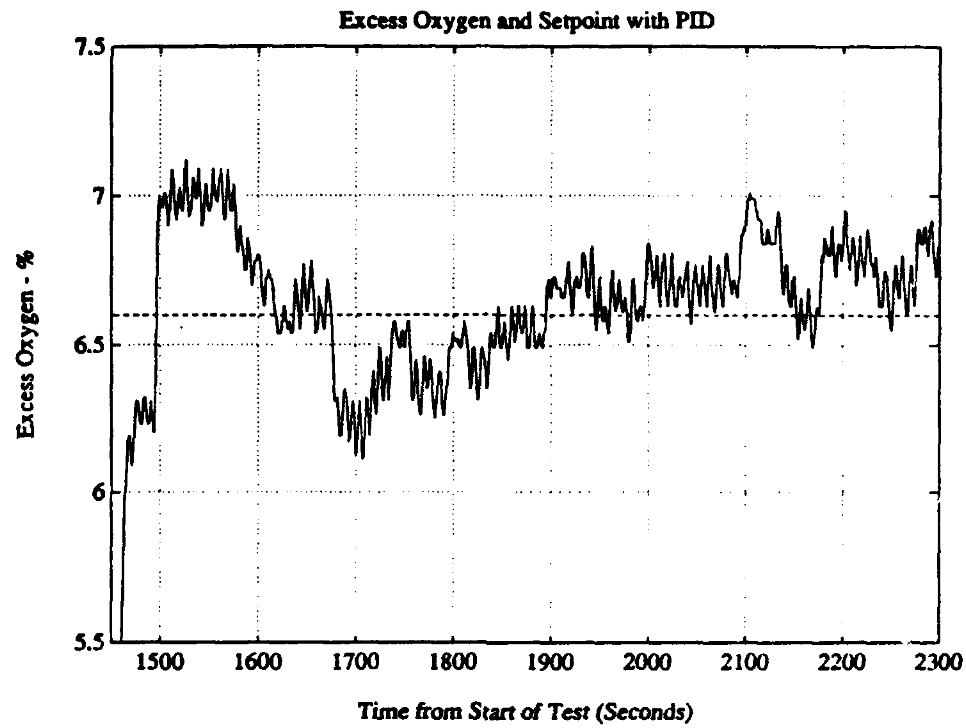


Figure 47. Step Response of Data Batch 7001 to 7500 Seconds of O2/FF; Delay=18 Seconds.



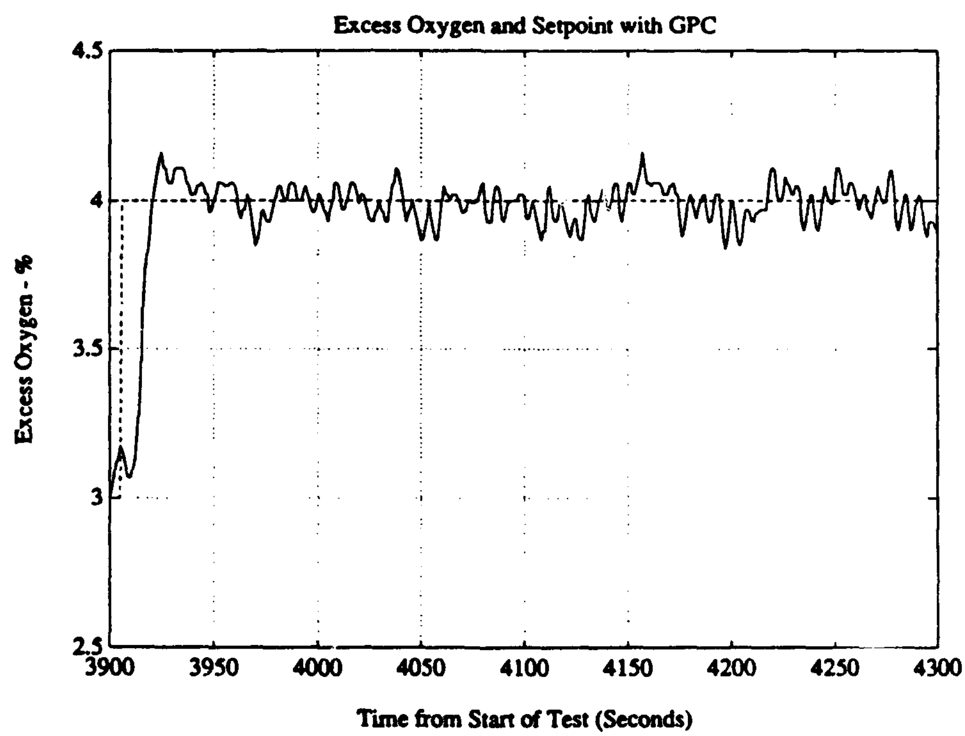


Figure 48. Comparison of the Step Response of PID and GPC for the Excess Oxygen Loop.

5 BOILER COMBUSTION EFFICIENCY CALCULATIONS

Combustion efficiency measurement is a very important issue, because it is an index directly indicative of boiler performance. It will be very helpful for the operator and/or control strategy in making adjustments to maintain optimal boiler productivity. Boiler combustion efficiency is defined as the ratio of the heat transferred to the water/steam side compared to the higher heating value of the fuel. The American Society of Mechanical Engineers (ASME) has established the most widely accepted two methods for boiler efficiency testing, the *input/output method* and the *heat loss method*, which are obtained through a standardized calculation procedure as described in the American Society of Mechanical Engineers (ASME) Performance Test Code (PTC) 4.1. (1965). It is generally agreed that the heat loss method provides more accurate results, while the input/output method is more easily understood and, for that reason, is often preferred.

The basic calculation for the input/output method is:

$$\text{Efficiency} = \frac{\text{Useful Heat Extracted}}{\text{Total Heat Input}} \times 100\% \qquad \text{[Eq 54]}$$

Where

Useful Heat Input = heat content of steam to the header plus the heat content of blowdown flow minus the heat content of boiler feedwater

Total Heat Input = heat released by the combustion.

The input/output ASME test is well suited for gas- and oil-fired boilers where the fuel firing rates are often accurately measured and the heating values of the fuels are known constants. It is advantageous to use because efficiency is calculated from fuel and steam flowmeters. This method often meshes well with existing plant instrumentation, while disadvantages are that: (1) the method is sensitive to flow measurement errors, and (2) all fuels must be accurately measured.

The Heat Loss Method is used to calculate individual losses, totalize them, and determine boiler efficiency by subtracting the total losses from 100 percent. The formula is:

$$\text{Efficiency} = \left(1 - \frac{\text{Heat Losses}}{\text{Total Heat Losses}}\right) \times 100\% \qquad \text{[Eq 55]}$$

This equation can be obtained from the input/output method simply by making the substitution: Total Heat Input = Useful Heat Extracted Plus Heat Losses. The heat loss method calculates boiler performance using an equation that is comparatively less sensitive to measurement errors in fuel flow or steam flow than that used in the input/output method. This is an advantage in solid fuel firing systems, where it is impossible to obtain a highly accurate fuel measurement. In the case of a boiler firing multiple fuels, a ratio of the fuel firing rates is required. Also, fewer on-line measurements are required and it is thought to usually provide a more accurate result. Its main disadvantage is that not all losses are computable such as losses due to radiation, which are estimated using ASME published tables.

The following six heat losses are those specified by the ASME PTC 4.1 (ASME 1965) abbreviated method in accordance with the heat loss method to calculate the heat losses.

1. Dry Gas Loss. This loss represents heat lost in dry flue gas. The amount of this loss depends on: (a) the stack gas temperature; (b) the amount of excess air used to fire the fuel; and (c) the composition of the flue gas.

2. Loss Due to Moisture from Burning. Hydrogen in hydrocarbon fuels forms water when burned. The heat of vaporization of this water is lost out of the stack.

3. Loss Due to Moisture in Fuel. If the fuel being burned contains any moisture, this water must be evaporated. The result is a heat loss.

4. Loss Due to Unburned Combustibles. Unburned or partially burned fuel represents a loss.

5. Losses from Radiation and Convection. These losses through the boiler structure are virtually impossible to measure. Instead, refer to ASME published tables to determine heat loss.

6. Unmeasured Losses. Many minor heat losses are lumped into this category, using an "engineering estimate."

The calculation procedure, either input/output method or heat loss method, is rather complicated due to the number of process variables that had to be measured and the number of calculations to be executed. For real-time application, it is better to find a simple and easy way to perform the boiler efficiency calculations.

Under the assumption that there was complete combustion and no water-vapor in the combustion air, it is possible to determine the combustion efficiency as a function of flue gas temperature and percentage O_2 in the dry flue gas. Based on the data provided by *Boiler Efficiency Improvement* (Maples and Dyer 1981), the combustion efficiency calculation models for different fuels can be obtained by using a regression method.

Efficiency of Natural Gas-Fired Boiler

$$\begin{aligned}\eta(\%) = &-80.6545 + 16.75665x_1 - 0.45138x_1^2 \\ &+ 29.4500x_2 - 1.25286x_2^2 - 1.44272x_1x_2 - 0.02769T\end{aligned}$$

where:

x_1 is the percentage O_2 in the dry flue gas
x_2 is the percentage CO_2 in the dry flue gas
T is the flue gas temperature.

Moreover, the percentage CO_2 in the dry flue gas can be estimated from the percentage O_2 in the dry flue gas via the chemical reaction balance in the combustion process. The percent CO_2 for the natural gas-fired unit is calculated by:

$$CO_2(\%) = 11.80121 - 0.56145\,O_2(\%)$$

The combustion efficiency estimation error is the difference between the calculated outputs and the table values provided in (Maples and Dyer 1981). Figure 49 shows the combustion efficiency estimation errors for the natural gas-fired boiler, where the oxygen is ranged from 0 to 10 percent, and the stack temperature is varied from 170 to 630 °F. It is interesting to note that the oxygen value is usually set at the region from 4 to 7 percent. In practice, for this region, the maximum estimation error is only about 0.5 percent; even the temperature varies from 170 to 630 °F. Figure 50 shows a comparison of the calculated percent CO_2 and the table values provided in Maples and Dyer. It appears that the estimated percent CO_2 values and the table values are very close.

Efficiency for No. 2 Oil-Fired Boiler

$$\begin{aligned}\eta(\%) &= 1202.0821 - 9.65112x_1 + 0.173219x_1^2 \\ &\quad -3.2171x_2 + 0.13508_2^2 + 0.590321x_1x_2 - 0.02684T\end{aligned}$$

where x_1, x_2, and T are defined similarly as in the case of Natural Gas. The percent CO_2 for the no. 2 Oil is:

$$CO_2(\%) = 15.61069 - 0.7425O_2(\%)$$

Figures 51 and 52 show the combustion efficiency estimation errors for a no. 2 oil-fired boiler and the comparison of the estimated percent CO_2 and the percent CO_2 table values from Maples and Dyer (1981).

Efficiency for a No. 6 Oil-Fired Boiler

$$\begin{aligned}\eta(\%) &= 23.29147 - 18.3246x_1 + 1.001373x_1^2 \\ &\quad + 14.6999x_2 - 0.61834x_2^2 + 0.820384x_1x_2 - 0.02719T\end{aligned}$$

where x_1, x_2, and T are defined similarly as in the case of natural gas. The percent CO_2 for the no. 6 oil-fired unit is obtained by:

$$CO_2(\%) = 16.49536 - 0.78476O_2(\%)$$

Figures 53 and 54 show the combustion efficiency estimation errors for a no. 6 oil-fired boiler and the comparison of the estimated percent CO_2 and the percent CO_2 table values from Maples and Dyer.

With the above equations, the boiler combustion efficiencies can be easily calculated for the natural gas, no. 2 oil and no. 6 oil fired boilers in real time. Figure 55 diagrams how the combustion efficiency is calculated.

The above combustion efficiency calculation algorithm has been implemented in the PC using the FIX software package. The database used to calculate the boiler efficiency is listed in Appendix F.

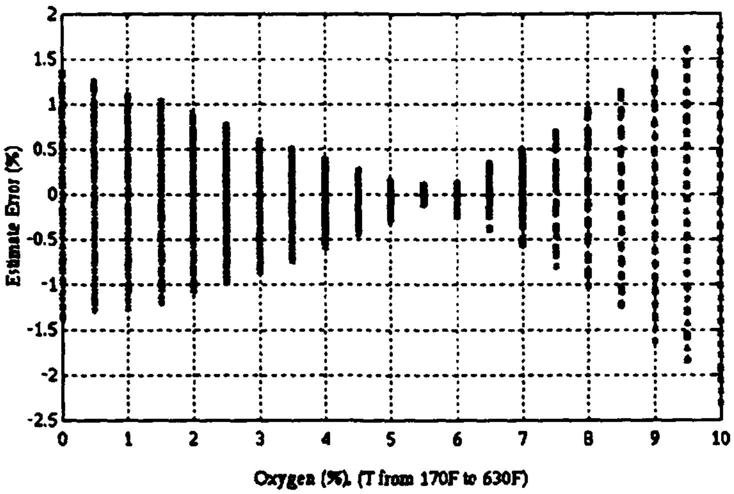


Figure 49. Combustion Efficiency Estimation Errors for Natural Gas-Fired Boiler.

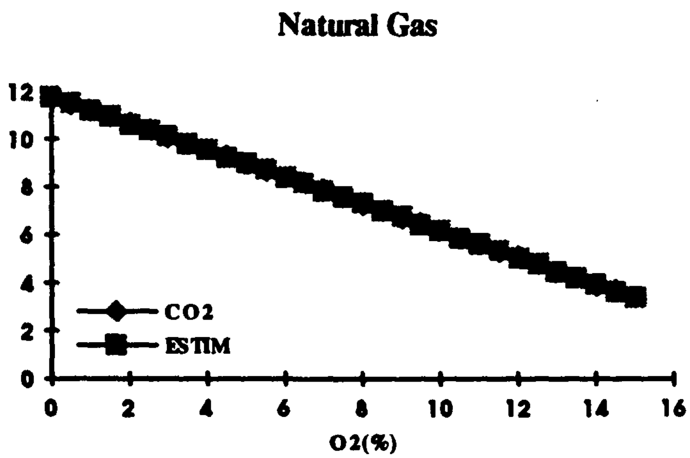


Figure 50. Estimated CO_2 Values for Natural Gas-Fired Boiler.

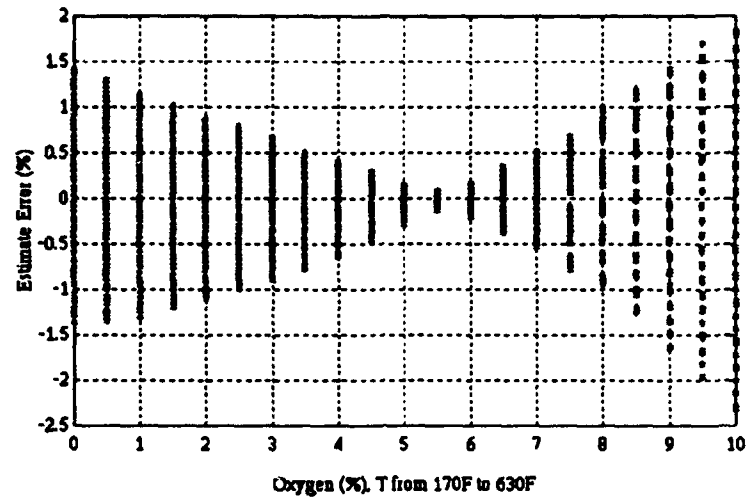


Figure 51. Combustion Efficiency Estimation Errors for #2 Oil-Fired Boiler.

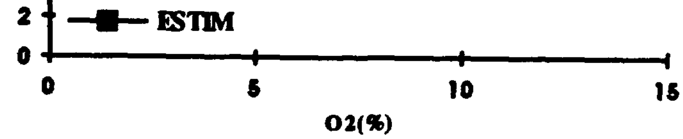


Figure 52. Estimated Percent CO_2 for #2 Oil-Fired Boiler.

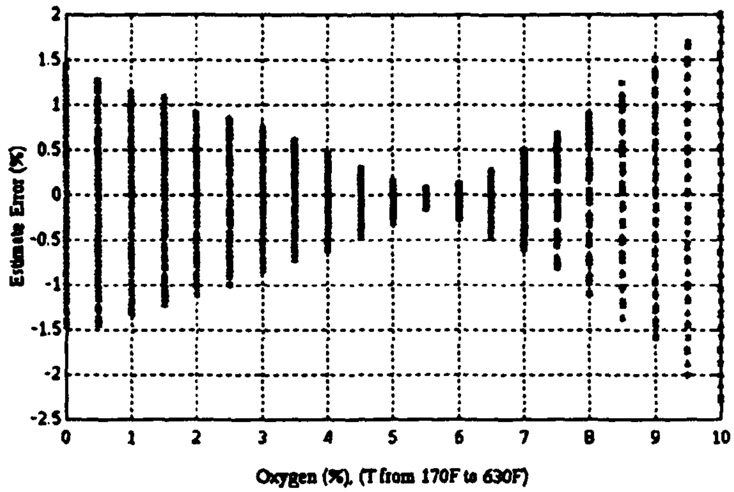


Figure 53. Combustion Efficiency Estimation Errors for #6 Oil-Fired Boiler.

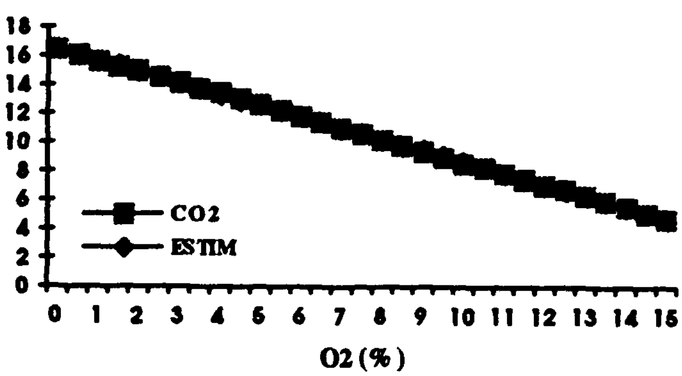


Figure 54. Estimated Percent CO_2 for #6 Oil-Fired Boiler.

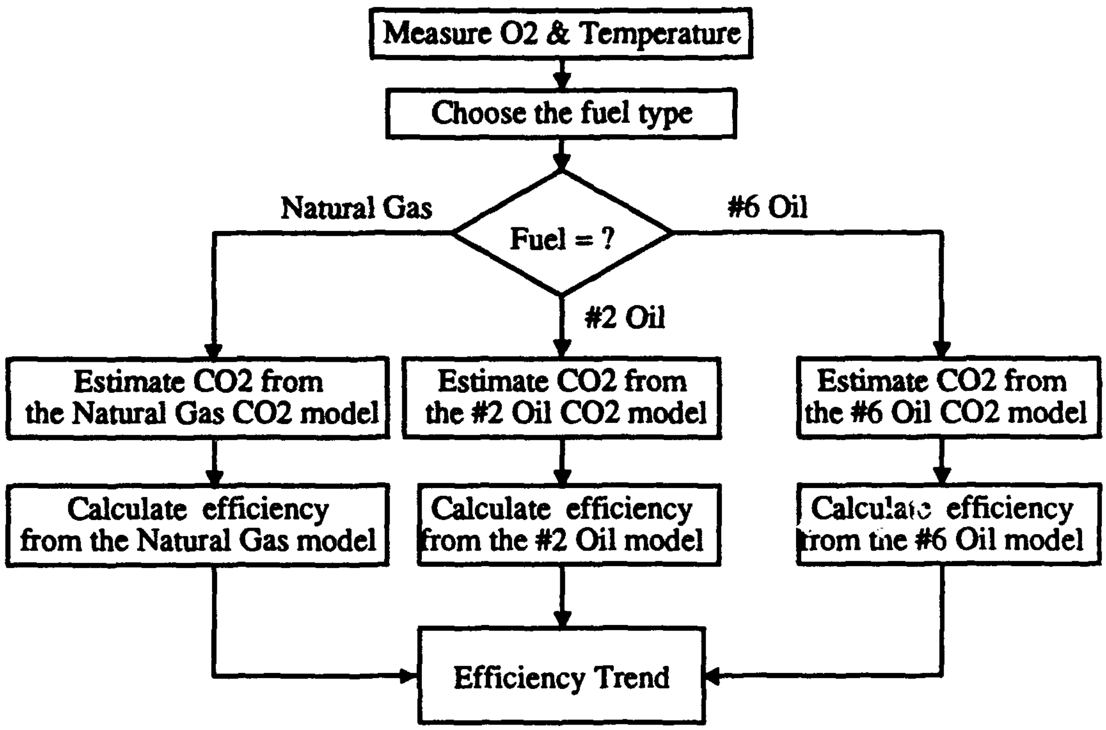


Figure 55. The Flow Chart of the Boiler Efficiency Calculations.

6 ROBUSTIFICATION OF THE IDENTIFIER

Real industrial systems can rarely be modeled to a high level of accuracy. In particular, a linear black-box type CARMA (Controlled Auto Regressive and Moving Average) or CARIMA (Controlled Auto-Regressive and Integrated Moving Average) model of low order (usually <= 4) as is normally assumed in self-tuning control, can only be expected to explain local system behavior. Real processes are fraught with unstructured disturbances and nonlinearities. A boiler is a highly nonlinear multivariable system with complex dynamics. When a low-order linear model is fitted to it, the results will depend critically on the frequency content of the input signal. Identification problems in the presence of noise and unmodeled dynamics have been attacked using two methods, slow sampling (Rohrs et al. 1984), and prefiltering (Åström and Wittenmark 1989; Mohtadi 1988).

Sampling Procedure and System Noise

The sampling procedure will automatically introduce a low-pass filter (Wahlberg 1990). Its break frequency w_{bf}(which is defined as the frequency at which the amplitude is 3 db below the maximum value of the amplitude), is proportional to the inverse of the sampling period T, i.e., $w_{bf} \approx 1/T$. The bandwidth of this low-pass filter tends to infinity with a converge rate 1/T as T tends to zero. Therefore, most of the high frequency noise and effects of high frequency unmodeled dynamics will be present in the sampled data set if a fast sampling rate is used.

A slow sampling rate is one of the possible solutions to the identification problem with unmodeled dynamics and noise, since the low-pass property of the slow sampling procedure will attenuate the gain at high frequencies. In the early development stages of the control algorithm, difficulties were experienced with simulations at fast sampling rates. Models converged to high frequency noise models. For this reason, a slow sampling approach was used initially. However, slow sampling rate lead to other identification problems. The deficiencies observed in the July 1991 tests appear to be with identification. Slow sampling rate seems to be the primary problem, because the estimator could not get enough real time information from the plant due to the data update rate being too low. This resulted in the parameter convergence speed being too low. In the case of the water level loop, with a sampling interval of 20 seconds, the parameters required over 40 minutes to converge. The tracking property of the identifier was poor with a low sampling rate; there was a heavy nonlinearity in the O_2 loop, and the identified model parameters needed to be changed with different operating points. Poor parameter tracking of the identifier lead to the loss of the plant model for the O_2 loop when the operating points sharply turned down (Figure 35, 6000 to 7000 seconds). Poor identification leads to poor control.

It now appears that better results can be achieved by using fast sampling with prefiltering. When a fast sampling rate is used, precautions must be taken to ensure that the frequency content of the input signal is concentrated in the frequency range where the simplified model is expected to fit well. This implies that the signals should be filtered before they enter into the parameter estimator.

Filter Types, Break Frequency, and Filter Rational

To attenuate the effect of disturbances wherever they are dominant, and to suppress the effects of unmodeled dynamics, a filter must be chosen. There are many types of filters available for this purpose. Classic filters include Butterworth, Bessel, Chebyshev, and Elliptic filters, any of which can be easily designed using standard filter design methods (Oppenheim and Schafer 1975; Parks and Burrus 1985). Also, there are many commercial filter design packages available in the market, for example, *Signal*

Processing Tool Box in MATLAB (PRO-MATLAB 1991). Low-pass or band-pass filters are often used to eliminate noise. In general, the type of filter used is not critical. The important parameter is the proper choice of the break frequency of the filters, since if the input is not a band limited signal, the filter will suppress useful information at the same time as the noise is reduced. The filter must reduce the effects of the external disturbances and those of unmodeled dynamics, thereby preventing sudden variations in the estimated parameters. As a rule of thumb, the lower break frequency w_{bl} should be at least 1 decade below the desired closed loop gain crossover frequency (the frequency at which the amplitude ratio of the control loops is zero db), and further, the upper break frequency w_{bh} should be about 2 to 10 times the gain crossover frequency. If w_{bh} is too high, the estimator may *still attempt to fit the model to high frequency* content noise and plant dynamics.

Normalization

In addition to slow sampling and prefiltering methods mentioned above, there are some other methods, such as normalization, data scaling, dead-zone, projection, etc., to robustify the identifier when noise and unmodeled dynamics are present. It is well known that, in practical applications of least-squares estimation, poor data often lead to numerical problems. This is especially true if the excitation of the process is poor. Normalization methods are used in this algorithm to improve the numerical quality of the computations and provide better estimation convergence (Middleton and Goodwin 1990; Sripada and Fisher 1987). The normalization method normalizes the regressor vector F(t), which is composed of input and output data sets $\{u(.)\}$ and $\{y(.)\}$, by the normalization factor $n(t) = \max(1, \|F(t)\|)$ so that all the elements of $F_n(t) = F(t)/n(t) < 1$. The normalization reduces the problem of unbounded disturbances to a problem of bounded disturbances.

Time Delay and Model Order

Determination of the correct time delay is an important issue in the modeling procedure. For boiler systems, time delay usually varies with different operating regimes. A fixed time delay for the whole operating range is not possible. One solution to this problem is the use of a higher order model to allow accommodation to varying time delay. For example, the water level loop basically behaves as an integrator with nonminimum phase response, so a first-order model could be used. To accommodate the varying time delay, a third-order model was used to model this loop. Simulation results show that with such a model the system could adapt to a fixed structural time delay varying from 10 to 23 seconds. But if the time delay were overestimated further, the step response of this loop will become unreasonable, such as when the time delay was set equal to 24 seconds. This suggests that when choosing a time delay, it is better to choose too short a delay rather than too long a delay. A higher order model must be used to increase the flexibility so that a varying time delay can be accommodated. In our simulation, when a lower order model was used, it seems that the plant still would be able to accommodate the time delay varying from 10 to 23 seconds.

7 FAULT TOLERANCE AND DIAGNOSTICS

The purpose of the fault tolerance system is to provide gracefully degrading performance under the deterioration and failure of both hardware and software system components, and/or incorrect operator actions, and under such conditions, to indicate the faulty component. Here we only consider the simplest fault tolerant features that could be implemented in DMACS supervisory software.

Typical failure modes for boiler control systems include:

1. Actuator failure, e.g., actuator sticking, motor burnout, and diaphragm leakage in pneumatic actuators.

2. Sensor (transmitter) failure or corruption of sensor (pressure,water level, oxygen) reading.

3. Deterioration of fuel quality, e.g., dilution of natural gas with additives, dilution of oil with water in oil-fired boilers, burner clogging.

4. Refractory breakdown that exposes boiler walls to damaging temperatures.

5. Leakage of pressurized water or steam from sight glass or superheatei tubes.

6. Controller failure (software and/or hardware).

Procedures listed below in the order corresponding to failure modes given above addressed each failure type.

1. Actuator deterioration is addressed by closing the local loop around each actuator (implemented in μMac code), which ensures faithful execution of the control signal under significant changes in actuator characteristics. The performance of these local loops is continually monitored, and when the performance measure (e.g., ITAE criterion, the Integral of Time Multiplied by Absolute Error Criterion (Dorf 1989):

$$I = \int_0^T t|e(t)|dt \qquad \text{[Eq 56]}$$

exceeds a chosen threshold, the actuator-specific failure signal is displayed on screen indicating that actuator performance is unacceptable.

2. Sensor failure is addressed by checking if sensor readings stay within the physically meaningful bounds. If bounds are exceeded, the sensor-specific failure signal is displayed. The graceful degradation of sensor readings could be carried out by combining the main sensor data with the data from a redundant set of sensors to arrive at compromise sensor reading. A plant model could be used to predict outputs for known inputs and compare them to actual output readings for discrepancy. If the discrepancy exceeds a threshold, a failure signal is displayed.

3. Fuel quality and burner clogging are monitored by periodically comparing normal projected boiler efficiency with the observed efficiency both computed on-line. A second technique involves checking for excessive fuel demand for the current setpoint.

4. Refractory breakdown can be detected by the discrepancy between nominal and actual air/fuel demand for the maintenance of a particular setpoint.

5. Leakage is detected by the pressure loss under nominal or excessive fuel/air consumption for a particular setpoint.

6. Controller failure is determined by comparing controller output with baseline (normal operation) for a given input and setpoint using some (e.g., ITAE) performance criterion. If an adaptive algorithm fails, control is switched to a PID controller in DMACS, which provides acceptable performance.

Identifiers in the μMac code provide information for pattern recognition, which could map parameter changes into a variety of failure modes and requests for maintenance for various subsystems. This aspect should be explored further.

8 CONCLUSIONS AND RECOMMENDATIONS

This study developed a prototype Central Energy Plant Adaptive Control System using a microcomputer in conjunction with third-party control software to collect data, perform advanced control functions, generate reports, and to serve as an alarm and an operator interface.

The prototype boiler control system was successfully field tested in July 1991 at the University of Illinois Abbbot Power Plant. The system demonstrated its ability to control steam pressure and stack oxygen levels significantly better than that of the conventional PID system. While the water level control loop and the excess oxygen loop showed deficiencies during abrupt steam demand reduction, these occurrences may have been caused by the slow sampling rate used during the test period. Identification protection schemes were implemented to permit higher sampling rates, and it is anticipated that the reconfigured controller will perform well under all conditions (Chapter 4).

During the field test, the controller hardware was critically examined and improvements were identified and made. Automatic startup and shutdown procedures were implemented, and possible self-diagnostic techniques were explored (Chapter 3).

The great advantage of the GPC system lies in its control loops, which exhibit significant non-linearities over the entire operating range. It is recommended that a longer term field demonstration be done to apply GPC techniques to an operating unit. Such a demonstration would ideally be performed in a location staffed by experienced operators and maintenance personnel. The project should exploit the knowledge and experience of academic developers, control system manufacturers, and a consulting engineer to develop a practical demonstration project of this promising technology.

REFERENCES

μMAC-6000: Hardware Technical Reference Manual (Analog Devices, Inc., 1988).

μMAC-6000: Operation Manual (Analog Devices, Inc., 1987).

AMTECH SERVICES, "Boilers and Boiler Controls," *Power Plant Training Seminar Notes* (Indianapolis, IN, 1990).

Åström, K.J., and K. Eklund, "A Simple Non-Linear Drum Boiler Model," *Int. J. Control*, Vol 22, No. 5 (1975), pp 739-740.

Åström, K.J. and K. Eklund, "A Simplified Non-Linear Model of a Drum Boiler-Turbine Unit," *Int. J. Control*, Vol 16, No. 1 (1972), pp 145-169.

Åström, K.J., and R.D. Bell, *Dynamic Models for Boiler-Turbine-Alternator Units: Data Logs and Parameter Estimation for a 160 MW Unit*, Report LUTFD2/(TFRT-3192)/1-137/(1987) (Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1987).

Åström, K.J., and R.D. Bell, "Simple Drum-Boiler Models," *Proceedings of the IFAC Symposium Power Systems, Modelling and Control Applications* (Brussels, Belgium, 1988).

Åström, K.J., and T. Hägglund, "Automatic Tuning of Simple Regulators With Specifications on Phase and Amplitude Margins," *Automatica*, No. 20 (1984), pp 645-651.

Åström, K.J., and B. Wittenmark, *Con.,uter Controlled System—Theory and Design* (Prentice-Hall, 1984).

Åström, K.J., and B. Wittenmark, *Adaptive Control* (Addison-Wesley, 1989).

REFERENCES (Cont'd)

Bell, R.D., and K.J. Åström, *A Low Order Nonlinear Dynamic Model for Drum Boiler-Turbine Alternator Units*, Report TFRT-7162 (Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1987).

Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation* (Academic Press, 1977).

Bitmead, R.R., M. Gevers, and V. Wertz, *Adaptive Optimal Control—The Thinking Man's GPC* (Prentice-Hall, 1990).

Chawdry, P.K. and B.W. Hogg, "Identification of Boiler Models," *IEEE Proc.* Vol 136, pt. D, No. 5 (1989), pp 261-271.

Chawdry, B.K., and B.W. Hogg, "Identification of Boiler Models," *IEEE Proc. D*, Vol 136, No. 5 (1989).

Clarke, D.W., C. Mohtadi, and P.S. Tuffs, "Generalized Predictive Control: Parts I and II," *Automatica*, Vol 23 (1987), pp 137-160.

Clarke, D.W., "Application of Generalized Predictive Control to Industrial Processes," *IEEE Control System Magazine*, Vol 8 (1988), pp 49-55.

Clarke, D.W., and T.T.C. Tsang, "Generalized Predictive Control With Input Constraints," *IEEE Procedings*, Vol 135 (1988), pp 451-460.

Dorf, R.C., *Modern Control Systems*, 5th ed. (Addison-Wesley, 1989).

Dukelow, S.G., *The Control of Boilers* (ISA Press, 1986).

Favier, G., "Self-Tuning Long Range Predictive Controller," *Proceedings of the 10th IFAC World Congress*, Vol 10 (Munich, 1987).

FIX DMACS User's Guide, Version 2.1 (Intellution, Inc., Norwood, MA, 1991).

GE Fanuc Automation, *Automation News* Vol 4, No. 2 (Spring 1993).

Goodwin, G.C., and R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis* (Academic Press, 1977).

Harris, C.J., and S.A. Billings, ed., *Self-Tuning and Adaptive Control: Theory and Applications* (Peter Peregrinus, 1985).

Lambert, E.P., *Process Control Applications of Long-Range Prediction*, Ph.D. Thesis (Oxford University, 1987).

Ljung, L., *System Identification Toolbox for Use With MATLABTM™—User's Guide* (The MathWorks, Inc., 1991).

Ljung, L., and T. Åström, *Theory and Practice of Recursive Identification* (The MIT Press, 1987).

Ljung, L., *System Identification-Theory for the User* (Prentice-Hall, 1987).

Maples, G., and D.F. Dyer, *Boiler Efficiency Improvement*, 3d ed. (The Boiler Efficiency Institute, Alabama, 1981).

McDonald, J.P., H.G. Kwatney, and J.H. Spare, "A Nonlinear Model for Reheat Boiler-Turbine-Generator Systems, Parts I and II," *Proc. JACC* (St. Louis, MO, 1971), pp 219-236.

Middleton, R.H., and G.C. Goodwin, *Digital Control and Estimation—A Unified Approach* (Prentice-Hall, 1990).

Mohtadi, C., "On the Role of Prefiltering in Parameter Estimation and Control," *Workshop on Adaptive Control Strategies for Industrial Use* (Lodge Kananaskis, Alberta, Canada, 20-22 June 1988).

Morton, A.J., and P.H. Price, "The Controllability of Steam Output, Pressure and Water Level in Drum Boilers," *Proc. I. Mech. Eng.* (1977), p 75-84.

NFPA 85A, *Standard for Prevention of Furnace Explosions in Fuel Oil and Natural Gas-Fired Single Burner Boiler-Furnaces* (National Fire Protection Association [NFPA], 1987).

REFERENCES (Cont'd)

NFPA 85B, *Standard for Prevention of Furnace Explosions in Natural Gas-Fired Multiple Burner Boiler-Furnaces* (NFPA, 1989).

NFPA 85G, *Standard for Prevention of Furnace Implosions in Multiple Burner Boiler-Furnaces* (NFPA, 1987).

Oppenheim, A.V., and R.W. Schafer, *Digital Signal Processing* (Prentice-Hall, 1975).

Parks, T.W., and C.S. Burrus, *Digital Filter Design* (John Wiley & Sons, 1987).

PRO-MATLAB User's Guide (The MathWorks, Inc., South Natick, MA, 1991).

PTC 4.1, *Performance Test Code Steam Generating Units* (The American Society of Mechanical Engineers, 1965).

Rohrs, C.E., M. Athans, L. Valavani, and G. Stein, "Some Design Guidelines for Discrete-Time Adaptive Controllers," *Automatica*, Vol 24 (1984), pp 653-660.

Sripada, N.R., and D.G. Fisher, "Improved Least Squares Identification," *Int. J. Control*, Vol 46, No. 6 (1987), pp 1889-1913.

THE FIX User's Guide, Version 3.0 (Intellution, Inc., Norwood, MA, 1990).

Wahlberg, B., "The Effects of Rapid Sampling in System Identification," *Automatica*, Vol 26, No. 1 (1990), pp 167-170.

Warwick, K., ed., *Implementation of Self-Tuning Controllers* (Peter Peregrinus, 1988).

Wilde, Douglas J., and C.J. Beighter, *Foundations of Optimization* (Prentice-Hall, Inc., 1967).

APPENDIX A: Plan for July Experiment: Closed Loop Control System Testing on Abbott Boiler No. 2 (Revised 9 July 1991)

Overall Experimental Procedure (GPC Test—Wednesday, 10 July)

1. Operators should set the oxygen for boiler No. 2 at a moderately high value, and stabilize the drum level with moderate boiler load. Other boilers on the header may be on automatic during the inner loop tuning procedure, but all M/A stations for boiler No. 2 should be on manual at this time.

2. Record the current time (use the AT's clock) and start the boiler control program (GPC version) on the Micromac.

3. Check the PC's hard disks for sufficient space to store data.

4. Startup the FIX on the PC.
   a. Start historical trend collect running in the background.
   b. Enter view GPC1.

5. Execute a bumpless transfer to micromac control (Check that all M/A stations are on manual. Switch the selector to "Mac." Match the command signals and switch the gas, air and water flow M/A stations to "Auto").

6. Execute the Inner Loop Startup Procedure detailed below. At this point, the actuator controllers are in place on each subsystem.

7. Execute the Outer Loop Startup Procedure (GPC case) detailed below. At this point, both the actuator controllers and process controllers are in place on each subsystem.

8. Record the parameter values for each loop.

9. Turn on the PC data log.

10. Put each subsystem through a sequence of small process setpoint changes to record its performance.

11. Apply a sequence of small steam demand disturbances by changing the fuel setting on boiler No. 3. Record the fuel settings.

12. Return each subsystem to mode 4.

13. Turn off the PC data log and save the file obtained. Gradually shift the distribution of load by changing the fuel setting on boiler No. 3 so as to bring boiler No. 2 into a different operating regime. Manually control actuator setpoints from Fix to maintain pressure, oxygen and drum level near desired values.

14. Reset the covariances on the three outer loops.

15. Re-execute the Outer Loop Startup Procedure (GPC case) detailed below. At this point, both the actuator controllers and process controllers are in place on each subsystem.

16. Record the parameter values for each loop.

17. Turn on the PC data log.

18. Put each subsystem through a sequence of small process setpoint changes to record its tracking performance.

19. Apply a sequence of small steam demand disturbances by changing the fuel setting on boiler No. 3. Record the fuel settings.

20. Gradually return the steam load back to its original distribution while leaving the outer loop controllers at their current settings, note whether good performance is maintained.

21. Put each subsystem through a sequence of small process setpoint changes to record its tracking performance.

22. Apply a sequence of small steam demand disturbances by changing the fuel to record its tracking performance.

Overall Experimental Procedure (PID Test—Thursday, 11 July)

1. Operators should set the oxygen for boiler No. 2 at a moderately high value, and stabilize the drum level with load on boiler No. 2 the same as for the GPC test. Other boilers on the header may be in automatic during the inner loop tuning procedure, but all M/A stations for boiler No. 2 should be on manual at this time.

2. Record the current time (use the AT's clock) and start the boiler control program (PID version) on the Micromac.

3. Check the PC's hard disks for sufficient space to store data.

4. Startup the FIX on the PC.

   a. Start historical trend collect running in the background.

   b. Enter view PID1.

5. Execute a bumpless transfer to micromac control (Check that all M/A stations are in manual. Switch the selector to "Mac." Match the command signals and switch the gas, air and water flow M/A stations to "Auto").

6. Execute the Inner Loop Startup Procedure detailed below. At this point, the actuator controllers are in place on each subsystem.

7. Execute the Outer Loop Startup Procedure (PID case) detailed below. At this point, both the actuator controllers and process controllers are in place on each subsystem.

8. Turn on the PC data log.

9. Put each subsystem through the same sequence of small process setpoint changes as was used in the GPC experiment.

10. Apply a sequence of small steam demand disturbances by changing the fuel setting on boiler No. 3 in the same sequence as was used in the GPC experiment (if possible).

11. Return each subsystem to mode 4.

12. Turn off the PC data log and save the file obtained.

13. Gradually shift the distribution of load by changing the fuel setting on boiler No. 3 in the same sequence as was used in the GPC experiment (if possible) so as to bring boiler No. 2 into a different operating regime (if possible, to the same regime as was used in the GPC experiment). Manually control the actuator setpoints from the FIX to maintain pressure, oxygen, and drum level near desired values.

14. Re-execute the Outer Loop Startup Procedure (PID case) detailed below. At this point, both the actuator controllers and process controllers are in place on each subsystem.

15. Turn on the PC data log.

16. Put each subsystem through a sequence of small process setpoint changes to record its tracking performance.

17. Apply a sequence of small steam demand disturbances by changing the fuel setting on boiler No. 3 in the same sequence as was used in the GPC experiment (if possible).

18. Gradually return the steam load back to its original distribution while leaving the outer loop controllers at their current settings. Note whether good performance is maintained.

19. Put each subsystem through a sequence of small process setpoint changes to record its tracking performance.

20. Apply a sequence of small steam demand disturbances by changing the fuel setting on boiler No. 3 in the same sequence as was used in the GPC experiment (if possible).

Inner Loop Startup Procedure—Using PID's on All Actuator Loops

1. Run the batch file "stpresp" from the compaq. The compaq will begin logging all I/O data via Kermit.

2. Step the first command (fuel flow, air flow, or water flow) by 4 percent in a safe I/O data via Kermit.

3. Wait for the associated measured process input flow to settle.

4. Step the command back to its original value.

5. Stop the data logging on compaq. This will automatically start Matlab, which runs an m-file and permits a view of the plots for delay and maximum slope, and computes initial estimates for the PID parameters via the Ziegler-Nichols rules.

6. Compare with plots and gains obtained off-line.

7. Exit matlab on the compaq.

8. If the step response data is satisfactory, get a hard copy:
   a. "print stpresp"
   b. "graph matlab".

9. Save the step response data, e.g., "copy stpresp.mat data\ffstep1.mat".

10. At the FIX keyboard, enter initial PID gains—either from step response program or off-line tuning, but cut the overall gain by a factor of 10 to ensure stability.

11. Turn on the actuator controller (switch to mode 3—inner loop control mode).

12. When the measured process input reaches steady state, run the batch file "actresp" from the compaq. The compaq will begin logging all I/O data via Kermit.

13. Step the first setpoint by 5 percent in a safe direction.

14. Wait for the associated measured process input flow to settle.

15. Step the setpoint back to its original value.

16. Stop the data logging on the compaq. This will automatically start Matlab, which runs an m-file and permits a view of a plot of the closed-loop response, and computes the rise time and percent overshoot.

17. Compare with plots and performance measures obtained off-line.

18. Exit matlab on the compaq.

19. If the actuator response data is satisfactory, get a hard copy:
   a. "print actresp"
   b. "graph matlab".

20. Save the actuator response data, e.g., "copy actresp.mat data\ffresp1.mat".

21. From the FIX keyboard, adjust the PID gains.

22. When the measured process input again reaches steady state, re-run the batch file "actresp" from the compaq. The compaq will again begin logging all I/O data via Kermit. Repeat steps 17 to 26 until the closed-loop response is satisfactory.

23. Match the current command with the controller output and switch the current loop back into mode 2.

24. Repeat steps 6 to 28 for each inner loop (fuel flow, air flow, water flow).

Outer Loop Procedure (GPC Case)

Begin in with all subsystems in mode 3—inner loop control mode. At this time, boiler No. 3 is still in automatic (i.e., maintaining pressure).

1. Put boiler No. 3 on manual.

2. At the FIX, enter view GPC2 via the PgDn key.

3. Switch all subsystems to mode 4—outer loop identification mode.

4. Select the current loop to tune (e.g., drum level—loop 5).

5. Run the step response batch file for the current loop (e.g., "wlstep") from the compaq. The compaq will begin logging all I/O data via Kermit.

6. Step the actuator setpoint for the current loop (e.g., water flow setpoint) by 4 percent in a safe direction.

7. Wait for the associated measured process output (drum pressure, oxygen, drum level) to settle.

8. Step the actuator setpoint back to its original value.

9. Stop the data logging on the compaq. This will automatically start Matlab, which runs an m-file that permits us to view the plots for delay. Check the measured delay against the delay assumed in the boiler control program. Change the assumed delay if necessary.

10. Compare with step response plots obtained off-line.

11. Exit matlab on the compaq.

12. If the step response data is satisfactory, get a hard copy:
    a. e.g., "print wlstep"
    b. "graph matlab".

13. Save the step response data, e.g., "copy wlstep.mat data\wlstep1.mat".

14. Provide further excitation to actuator setpoints.

15. When the parameters for a loop converge (i.e., parameter values appear stable and give reasonable gains and rise times, prediction error is small, covariance mean diagonal is small), turn on the outer loop's controller (i.e., switch to mode 5—control mode).

16. When the measured process output reaches steady state, run the associated closed-loop response batch file (e.g., "wlresp") from the compaq. The compaq will begin logging all I/O data via Kermit.

17. Step the process setpoint by a reasonable amount (e.g., 5 psi for drum pressure, 2 percent for oxygen, 0.5 in. for drum level) in a safe direction.

18. Wait for the measured process output to settle.

19. Step the process setpoint back to its original value.

20. Stop the data logging on the compaq. This will automatically start Matlab, which runs an m-file that permits us to view a plot of the closed-loop response, and computes the rise time and percent overshoot.

21. Compare with plots and performance measures obtained off-line.

22. Exit matlab on the compaq.

23. If the outer loop response data is satisfactory, get a hard copy:

    a. e.g., "print wlresp"

    b. "graph matlab".

24. Save the actuator response data, e.g., "copy wlresp.mat data\wlresp1.mat".

25. If the response is unsatisfactory, return to mode 4 and repeat steps 14 to 23.

26. If the response is satisfactory, leave the current loop in mode 5.

27. Repeat the above procedure for the next outer loop (drum water level, drum pressure, oxygen).

Outer Loop Startup Procedure (PID Case)

Begin with all subsystems in mode 3—inner loop control mode. At this time, boiler No. 3 is still on automatic (i.e., maintaining pressure).

1. Put boiler No. 3 on manual.

2. At the FIX, enter view PID2 via the PgDn key.

3. Switch all subsystems to mode 4—outer loop identification mode.

4. Select the current loop to tune (e.g., drum level—loop 5).

5. Run the step response batch file for the current loop (e.g., "wlstep") from the compaq. The compaq will begin logging all I/O data via Kermit.

6. Step first actuator setpoint (fuel flow, air flow, or water flow) by 4 percent in a safe direction.

7. Wait for the associated measured process output (drum pressure, oxygen, drum level) to settle.

8. Step the actuator setpoint back to its original value.

9. Stop the data logging on compaq. This will automatically start Matlab, which runs an m-file that permits us to view the plots for delay and maximum slope, and computes initial estimates for the PID parameters via the Ziegler-Nichols rules.

10. Compare with plots and gains obtained off-line.

11. Exit matlab on the compaq.

12. If the step response data is satisfactory, get a hard copy:
    a. e.g., "print wlstep"
    b. "graph matlab".

13. Save the step response data, e.g., "copy wlstep.mat data\wlstep1.mat."

14. At the FIX keyboard, enter initial PID gains—either from step response program or off-line tuning, but cut the overall gain by a factor of 10 to ensure stability.

15. Turn on the outer-loop controller (switch to mode 5—full control mode).

16. When the measured process output reaches steady state, run the associated closed-loop response batch file (e.g., "wlresp") from the compaq. The compaq will begin logging all I/O data via Kermit.

17. Step the first setpoint by a reasonable amount (e.g., 5 psi for drum pressure, 2 percent for oxygen, 0.5 in. for drum level) in a safe direction.

18. Wait for the measured process output to settle.

19. Step the setpoint back to its original value.

20. Stop the data logging on the compaq. This will automatically start Matlab, which runs an m-file that permits us to view a plot of the closed-loop response, and computes the rise time and percent overshoot.

21. Compare with plots and performance measures obtained off-line.

22. Exit matlab on the compaq.

23. If the outer loop response data is satisfactory, get a hard copy:
    a. e.g., "print wlresp"
    b. "graph matlab".

24. Save the actuator response data, e.g., "copy wlresp.mat data\wlresp1.mat".

25. From the FIX keyboard, adjust the PID gains.

26. When the measured process input again reaches steady state, re-run the batch file "actresp" from the compaq. The compaq will again begin logging the loop's I/O data via Kermit. Repeat steps 17 to 26 until the closed-loop response is satisfactory.

27. Leave the current loop back in mode 5.

28. Repeat steps 6 to 28 for the next outer loop (drum pressure, oxygen, drum level).

APPENDIX B: PID-Controller Parameters Tuning Rules

In general, there are two basic methods for tuning PID-controller parameters: the transient-response method and the ultimative-sensitivity method (Åström and Wittenmark 1984). The continuous-time PID controller is often written in Laplace form as:

$$U(s) = K\left(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + T_D s/N}\right) E(s) \qquad \text{[Eq B1]}$$

Where:

U(s) and *E(s)* are the Laplace transforms of the controller output and the error signal, respectively
K is the proportional gain
T_I is the integral time
T_D is the derivative time.

In the controller there is a filter, with time constant T_D/N, for the derivative part. Also, constant N is often in the range 3 to 10.

The digital PID-controller is parameterized by approximating the derivative with a forward difference (Euler's method) as:

$$u(KT) = K_d\left[1 + \frac{T}{T_{id}(q-1)} + \frac{T_{dd}(q-1)}{T(q+\gamma)}\right] e(KT) \qquad \text{[Eq B2]}$$

Where:

u(kT) and *e(kT)* are the controller output and the error signal at time kT
k is the sampling numbers
T is the sampling period
K_d is the proportional gain
T_{id} and T_{dd} are the discrete-time equivalent to integral and derivative times
q is a difference operator
$\gamma = -\exp(-TN/T_d)$ is a constant.

The Transient-Response Method

To use the transient-response method, the steepest slope, R, and the delay time, L, are measured from a unit-step response of the open-loop system (Figure B1). The parameters for the PID-controller are then obtained from Table B1.

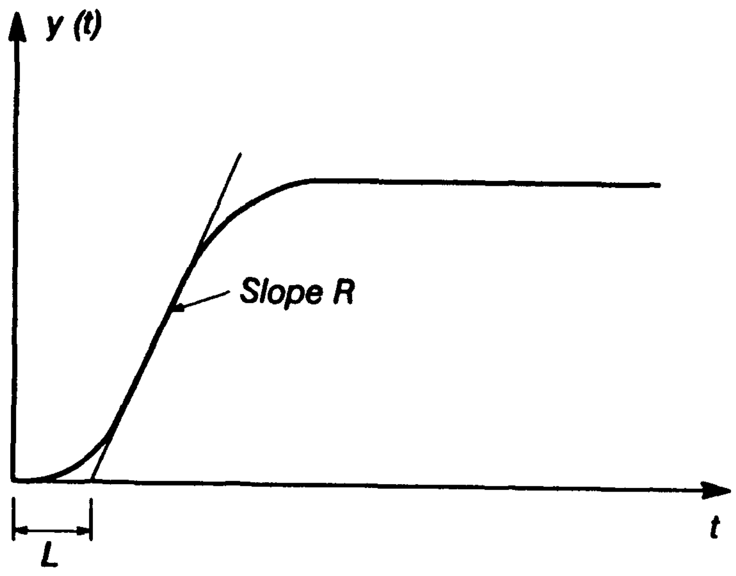


Figure B1. Measurement on the Open Loop Unit-Step Response of the Variables R and L for the Transient-Response Method.

Table B1

Controller Parameters When Using the Transient-Response Method

	Kd	Tid	Tdd
P	1/RL		
PI	0.9/RL	3L	
PID	1.2/RL	2L	0.5L

The Ultimative-Sensitivity Method

In the ultimative-sensitivity method, a P-controller is used first to control the system. The gain of the controller, K_{max}, and the period time, T_p, are measured when the closed-loop system is on the stability boundary. The parameters of the controller are then obtained from Table B2.

Table B2

Controller Parameters When Using the Ultimative-Sensitivity Method

	Kd	Tid	Tdd
P	$0.5K_{max}$		
PI	$0.45K_{max}$	$T_p/1.2$	
PID	$0.6K_{max}$	$T_p/2$	$T_p/8$

The tuning rules above should be used only as a first approximation. The final tuning usually has to be done manually. In general, the transient-response method is easy to use, since it only needs to measure the unit-step response of the open loop system. The ultimative-sensitivity method, however, requires experimentation on the closed loop system to find the maximum gain kmax and the oscillatory period time Tp . This is often undesirable in the industrial processes for safety reasons. Hence, the ultimative-sensitivity method is only used for processes with slow dynamics such as those that occur in industrial furnaces.

APPENDIX C: Database for Startup and Shutdown Procedures

TAG............. SS-MODE
DESC............ SELECTION OF STARTUP OR SHUTDOWN
DV-DEVICE....... AD6
HT-H/W OPTIONS.. MAC6000
IO-ADDR......... 1:0:I45
SC-SIG COND..... NONE
EL-LO EGU....... 0.00
EH-HI EGU....... 100.00
ET-EGU TAG......
PA-PLANT AREA... ALL
AE-ALM ENABLE... DISABLE
NX-NEXT BLK.....
LL-LO OP LIMIT.. 0.00
LH-HI OP LIMIT.. 100.00
LR-RATE LIMIT... 100.00
CS-COLD START... 0.00
WS-WARM START... NO
OR-OUT REVERSE.. NO

TAG............. PROMPT
DESC............ STATE OF PROMPT
DV-DEVICE....... AD6
HT-H/W OPTIONS.. MAC6000
IO-ADDR......... 1:0:I46
SC-SIG COND..... NONE
EL-LO EGU....... 0.00
EH-HI EGU....... 100.00
ET-EGU TAG......
PA-PLANT AREA... ALL
AE-ALM ENABLE... DISABLE
NX-NEXT BLK.....
LL-LO OP LIMIT.. 0.00
LH-HI OP LIMIT.. 100.00
LR-RATE LIMIT... 100.00
CS-COLD START... 0.00
WS-WARM START... NO
OR-OUT REVERSE.. NO

TAG............. FLAG
DESC............ PROMPT FLAG
DV-DEVICE....... AD6
HT-H/W OPTIONS.. MAC6000
IO-ADDR......... 1:0:I47
SC-SIG COND..... NONE
EL-LO EGU....... 0.00
EH-HI EGU....... 100.00

ET-EGU TAG......
PA-PLANT AREA... ALL
AE-ALM ENABLE... DISABLE
NX-NEXT BLK.....
LL-LO OP LIMIT.. 0.00
LH-HI OP LIMIT.. 100.00
LR-RATE LIMIT... 100.00
CS-COLD START... 5.00
WS-WARM START... NO
OR-OUT REVERSE.. NO

TAG.............	A_PROMPT
DESC............	ACHO VALUE OF PROMPT FROM UMAC
DV-DEVICE.......	AD6
HT-H/W OPTIONS..	MAC6000
IO-ADDR.........	1:0:I48
SC-SIG COND.....	NONE
EL-LO EGU.......	0.00
EH-HI EGU.......	100.00
ET-EGU TAG......	
SM-SMOOTHING....	0
IS-INIT SCAN....	ON
ST-SCAN TIME....	5
NX-NEXT BLK.....	
LL-LO LO ALM....	0.00
AL-LO ALARM.....	0.00
AH-HI ALARM.....	100.00
HH-HI HI ALM....	100.00
RC-ROC ALM......	0.00
DB-DEAD BAND....	0.20
PA-PLANT AREA...	ALL
AP-ALM PRI......	L
AE-ALM ENABLE...	ENABLE
IA-INIT A/M.....	AUTO

TAG.............	A_FLAG
DESC............	Acho value of FLAG from umac
DV-DEVICE.......	AD6
HT-H/W OPTIONS..	MAC6000
IO-ADDR.........	1:0:I49
SC-SIG COND.....	NONE
EL-LO EGU.......	0.00
EH-HI EGU.......	100.00
ET-EGU TAG......	
SM-SMOOTHING....	0
IS-INIT SCAN....	ON
ST-SCAN TIME....	5
NX-NEXT BLK.....	

LL-LO LO ALM....	0.00
AL-LO ALARM.....	0.00
AH-HI ALARM.....	100.00
HH-HI HI ALM....	100.00
RC-ROC ALM......	0.00
DB-DEAD BAND....	0.20
PA-PLANT AREA...	ALL
AP-ALM PRI......	L
AE-ALM ENABLE...	ENABLE
IA-INIT A/M.....	AUTO

TAG...........	MAIN
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	5
IS-INIT SCAN..	ON
IA-INIT A/M...	AUTO
DESC..	MAIN PROCEDURES FOR STARTUP & SHUTDOWN
00.	SETLIM 0.2000
01.	IF SS-MODE = 1.00 GOTO 4
02.	IF SS-MODE = 2.00 GOTO 10
03.	GOTO 1
04.	CALL STARTUP
05.	SETOUT PROMPT 28.00
06.	DELAY 10
07.	SETOUT SS-MODE 0.00
08.	GOTO 1
09.	NUL
10.	CALL SHUTDN
11.	END
12.	NUL
13.	NUL
14.	NUL
15.	NUL
16.	NUL
17.	NUL
18.	NUL
19.	NUL

TAG...........	STARTUP
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	STARTUP PROCEDURES
00.	CALL START1

01. SETOUT PROMPT 5.00
02. DELAY 10
03. WAITFOR A_PROMPT = 0.00
04. CALL START2
05. CALL START3
06. CALL START4
07. CALL START5
08. CALL START6
09. NUL
10. NUL
11. NUL
12. NUL
13. NUL
14. NUL
15. NUL
16. NUL
17. NUL
18. NUL
19. NUL

TAG...........	START1
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	SETUP WF, WL & AF ACTUATORS
00.	SETOUT WF-SET 50.00
01.	SETOUT WL-SET 0.00
02.	SETLIM 10.0000
03.	WAITFOR WF >= 50.00
04.	SETLIM 0.2000
05.	SETOUT PROMPT 1.00
06.	DELAY 10
07.	WAITFOR A_PROMPT = 0.00
08.	SETOUT PROMPT 2.00
09.	DELAY 10
10.	WAITFOR A_PROMPT = 0.00
11.	SETOUT AF-SET 40.00
12.	SETOUT PROMPT 3.00
13.	DELAY 10
14.	WAITFOR A_PROMPT = 0.00
15.	SETOUT PROMPT 4.00
16.	DELAY 10
17.	WAITFOR A_PROMPT = 0.00
18.	NUL
19.	NUL

TAG...........	START2
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	PREPROCEDURE FOR IGNITE
00.	SETOUT PROMPT 6.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	SETOUT PROMPT 7.00
04.	DELAY 10
05.	WAITFOR A_PROMPT = 0.00
06.	SETOUT PROMPT 8.00
07.	DELAY 10
08.	WAITFOR A_PROMPT = 0.00
09.	SETOUT PROMPT 9.00
10.	DELAY 10
11.	WAITFOR A_PROMPT = 0.00
12.	SETOUT PROMPT 10.00
13.	DELAY 10
14.	WAITFOR A_PROMPT = 0.00
15.	SETOUT PROMPT 11.00
16.	DELAY 10
17.	WAITFOR A_PROMPT = 0.00
18.	NUL
19.	NUL

TAG...........	START3
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	ATTEMPT TO IGNITE THE IGNITOR
00.	SETOUT PROMPT 12.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	DELAY 10
04.	IF A_FLAG = 1.00 GOTO 18
05.	DELAY 60
06.	SETOUT PROMPT 13.00
07.	DELAY 10
08.	WAITFOR A_PROMPT = 0.00
09.	SETOUT PROMPT 12.00
10.	DELAY 10
11.	WAITFOR A_PROMPT = 0.00
12.	DELAY 10

```
13.                 IF A_FLAG = 1.00 GOTO 18
14.                 SETOUT PROMPT 14.00
15.                 DELAY 10
16.                 WAITFOR A_PROMPT = 0.00
17.                 END
18.                 NUL
19.                 NUL


TAG...........      START4
PA-PLANT AREA.      ALL
AE-ALM ENAB...      ENABLE
AP-ALM PRI....      L
ST-SCAN TIME..      1
IS-INIT SCAN..      OFF
IA-INIT A/M...      AUTO
DESC..              ENGAGE FF CONTROL
00.                 SETOUT PROMPT 15.00
01.                 DELAY 10
02.                 WAITFOR A_PROMPT = 0.00
03.                 DELAY 10
04.                 IF A_FLAG = 1.00 GOTO 17
05.                 CALL START2
06.                 CALL START3
07.                 SETOUT PROMPT 15.00
08.                 DELAY 10
09.                 WAITFOR A_PROMPT = 0.00
10.                 DELAY 10
11.                 IF A_FLAG = 1.00 GOTO 17
12.                 DELAY 60
13.                 SETOUT PROMPT 16.00
14.                 DELAY 10
15.                 WAITFOR A_PROMPT = 0.00
16.                 END
17.                 NUL
18.                 NUL
19.                 NUL


TAG...........      START5
PA-PLANT AREA.      ALL
AE-ALM ENAB...      ENABLE
AP-ALM PRI....      L
ST-SCAN TIME..      1
IS-INIT SCAN..      OFF
IA-INIT A/M...      AUTO
DESC..              ENGAGE FF CONTROL
00.                 SETOUT PROMPT 17.00
01.                 DELAY 10
02.                 WAITFOR A_PROMPT = 0.00
03.                 SETOUT PROMPT 18.00
```

04.	DELAY 10
05.	WAITFOR A_PROMPT = 0.00
06.	SETOUT FF-SET FF
07.	SETOUT PROMPT 19.00
08.	DELAY 10
09.	WAITFOR A_PROMPT = 0.00
10.	SETOUT PROMPT 20.00
11.	DELAY 10
12.	WAITFOR A_PROMPT = 0.00
13.	SETOUT PROMPT 21.00
14.	DELAY 10
15.	WAITFOR A_PROMPT = 0.00
16.	SETOUT PROMPT 22.00
17.	DELAY 10
18.	WAITFOR A_PROMPT = 0.00
19.	NUL

TAG...........	START6
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	ENGADE OUTER LOOP CONTROL
00.	SETOUT PROMPT 23.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	SETOUT PROMPT 24.00
04.	DELAY 10
05.	WAITFOR A_PROMPT = 0.00
06.	SETOUT M-O2-SET O2
07.	SETOUT PROMPT 25.00
08.	DELAY 10
09.	WAITFOR A_PROMPT = 0.00
10.	SETOUT OVR-ENGA 0.00
11.	SETOUT PROMPT 26.00
12.	DELAY 10
13.	WAITFOR A_PROMPT = 0.00
14.	SETOUT DP-SET DP
15.	SETOUT PROMPT 27.00
16.	DELAY 10
17.	WAITFOR A_PROMPT = 0.00
18.	SETOUT DP-SET 320.00
19.	NUL

TAG...........	SHUTDN
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE

AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	SHUT DOWN PROCEDURE
00.	NUL
01.	CALL SHUT1
02.	CALL SHUT2
03.	CALL SHUT3
04.	END
05.	NUL
06.	NUL
07.	NUL
08.	NUL
09.	NUL
10.	NUL
11.	NUL
12.	NUL
13.	NUL
14.	NUL
15.	NUL
16.	NUL
17.	NUL
18.	NUL
19.	NUL

TAG...........	SHUT1
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	SHUT DOWN DP, O2 & FF LOOPS
00.	SETOUT PROMPT 30.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	SETOUT DP-SET 0.00
04.	DELAY 60
05.	SETOUT PROMPT 31.00
06.	DELAY 10
07.	WAITFOR A_PROMPT = 0.00
08.	SETOUT PROMPT 32.00
09.	DELAY 10
10.	WAITFOR A_PROMPT = 0.00
11.	SETOUT PROMPT 33.00
12.	DELAY 10
13.	WAITFOR A_PROMPT = 0.00
14.	SETOUT FF-SET 0.00
15.	DELAY 30

16.	NUL
17.	NUL
18.	NUL
19.	NUL

TAG...........	SHUT2
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	2
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	SHUT DOWN THE REST
00.	SETOUT PROMPT 34.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	SETOUT PROMPT 35.00
04.	DELAY 10
05.	WAITFOR A_PROMPT = 0.00
06.	SETOUT PROMPT 36.00
07.	DELAY 10
08.	WAITFOR A_PROMPT = 0.00
09.	SETOUT PROMPT 37.00
10.	DELAY 10
11.	WAITFOR A_PROMPT = 0.00
12.	SETOUT PROMPT 38.00
13.	DELAY 10
14.	WAITFOR A_PROMPT = 0.00
15.	NUL
16.	NUL
17.	NUL
18.	NUL
19.	NUL

TAG...........	SHUT3
PA-PLANT AREA.	ALL
AE-ALM ENAB...	ENABLE
AP-ALM PRI....	L
ST-SCAN TIME..	1
IS-INIT SCAN..	OFF
IA-INIT A/M...	AUTO
DESC..	SHUTDOWN WATER FLOW
00.	SETOUT PROMPT 39.00
01.	DELAY 10
02.	WAITFOR A_PROMPT = 0.00
03.	SETOUT PROMPT 40.00
04.	DELAY 10
05.	WAITFOR A_PROMPT = 0.00
06.	SETOUT WF-SET 0.00

```
07.    DELAY 30
08.    SETOUT PROMPT 41.00
09.    DELAY 10
10.    WAITFOR A_PROMPT = 0.00
11.    SETOUT PROMPT 42.00
12.    NUL
13.    NUL
14.    NUL
15.    NUL
16.    NUL
17.    NUL
18.    NUL
19.    NUL
```

APPENDIX D: FIX DMACS Graphic Environment for the Automatic Startup and Shutdown Procedures

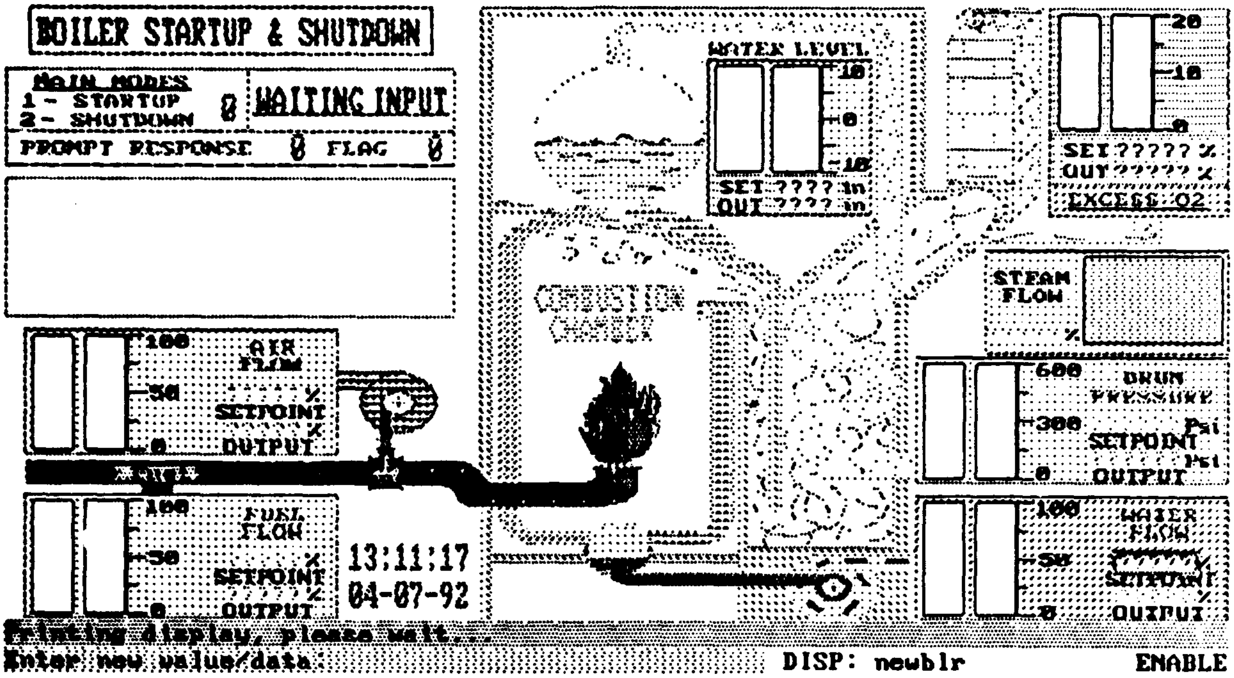


Figure D1. Boiler Startup and Shutdown.

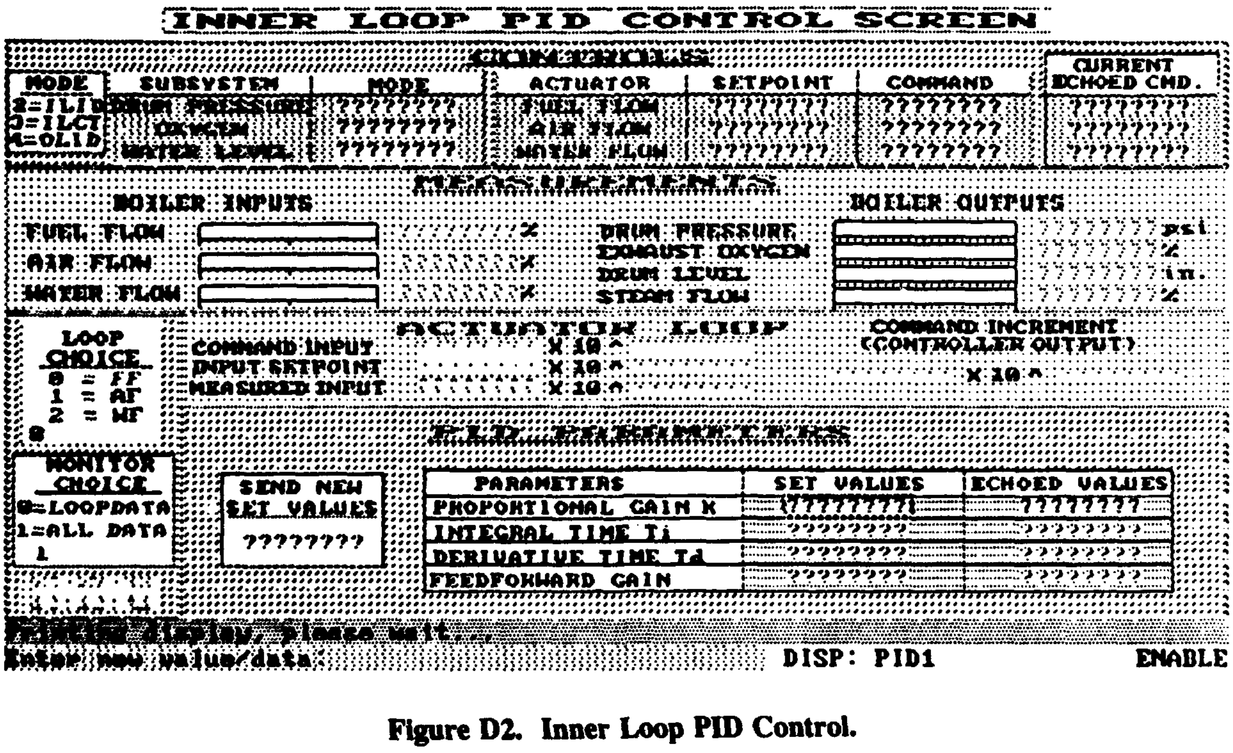


Figure D2. Inner Loop PID Control.

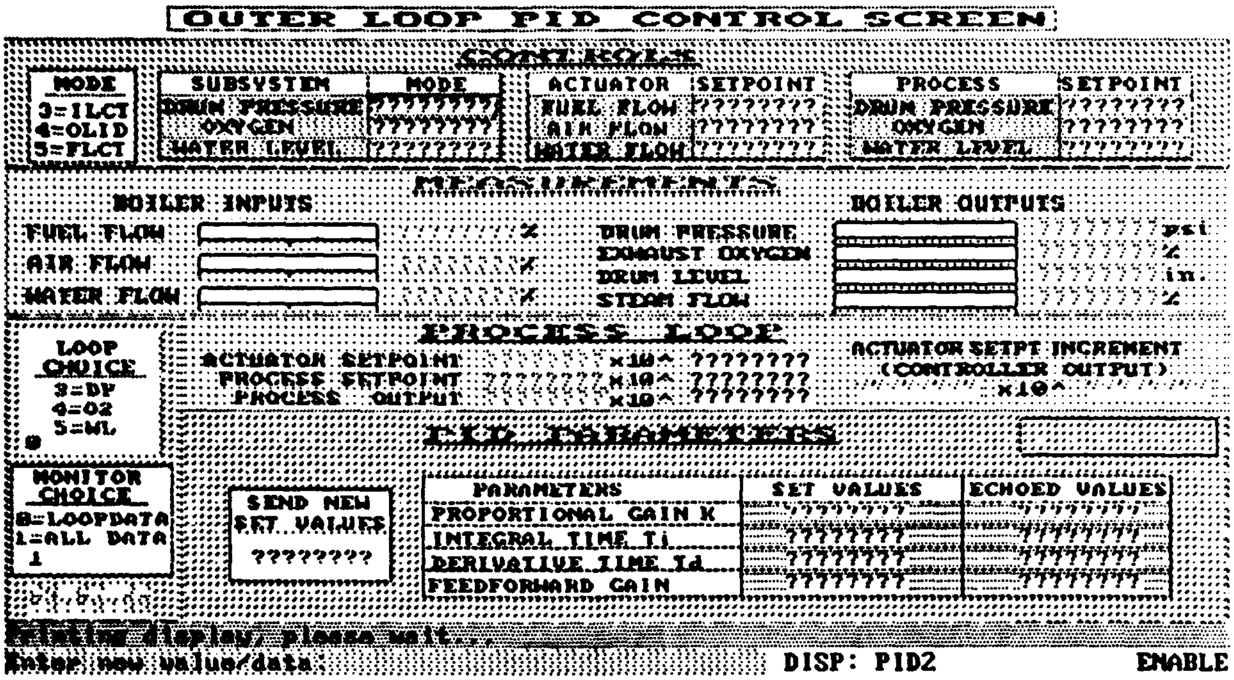


Figure D3. Outer Loop PID Control.

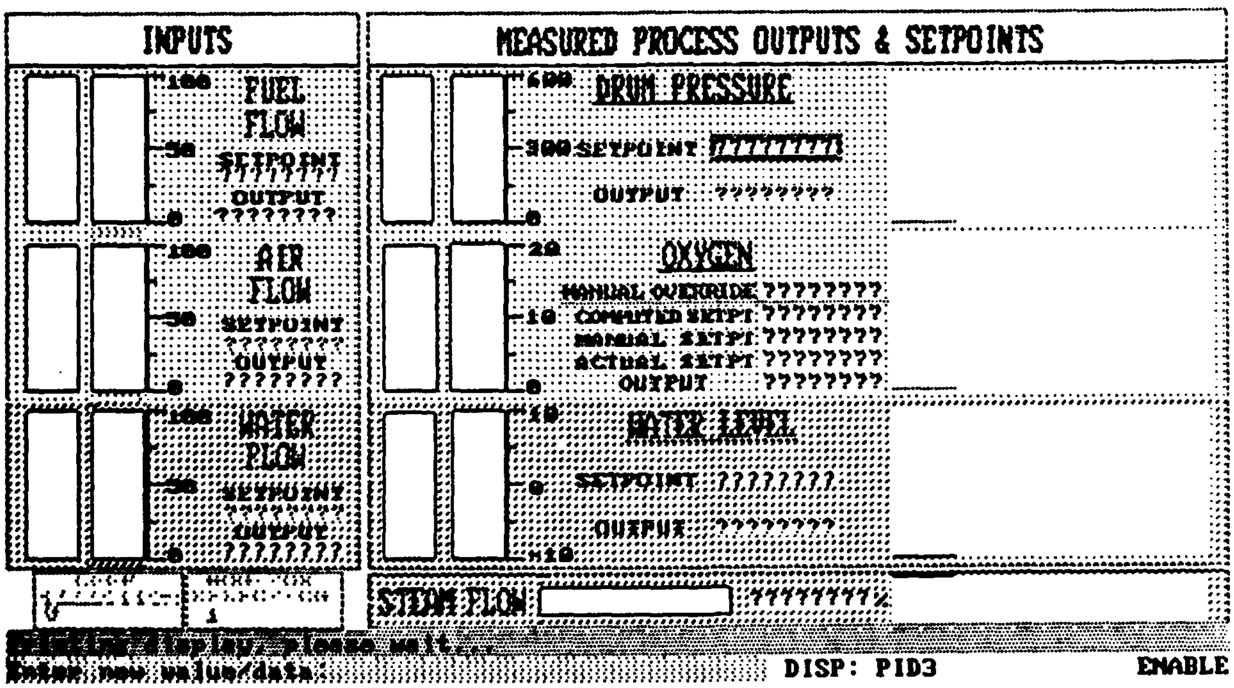


Figure D4. Output

APPENDIX E: Prompt Lists

PROMPTX:BAD PROMPT!
PROMPT0:JOB DONE!
PROMPT1:PRESS PgDn TO ENGAGE WF & WL CONTROLLERS
PROMPT2:OPEN DAMPER, OPEN FLOW PATH FROM AIR TO STACK
PROMPT3:PRESS PgDn TO ENGAGE AF CONTROLLER
PROMPT4:PRECHARGE THE BURNER HEADER WITH GAS, AND CHECK FOR LEAKS
PROMPT5:VERIFY THAT DAMPER AND BURNER REGISTER ARE IN PURGE POSITION
PROMPT6:PERFORM A UNIT PURGE
PROMPT7:CLOSE THE MAIN FUEL CONTROL VALVE, OPEN THE SAFETY SHUTOFF VALVE
PROMPT8:ESTABLISH HEADER PRESSURE FOR BURNER LIGHTOFF
PROMPT9:ESTABLISH IGNITOR PRESSURE FOR IGNITION
PROMPT10:SET AIR REGISTER TO LIGHTOFF POSITION
PROMPT11:IGNITE THE IGNITOR
PROMPT12:IF A FLAME IS NOT ESTABLISHED. CLOSE THE IGNITOR SAFETY VALVE. ELSE, SET FLAG = 1
PROMPT13:RETRY IGNITE THE IGNITOR
PROMPT14:SECOND IGNITION FAILED! THE STARTUP PROCEDURE WILL HALT!
PROMPT15:IF THE IGNITOR IS OPERATING CORRECTLY, SET FLAG=1 & SUPPLY FUEL TO THE BURNER! ELSE, SET FLAG=0
PROMPT16:SECOND FAILURE TO LIGHTOFF, THE STARTUP PROCEDURE WILL HALT!
PROMPT17:HAS A STABLE FLAME BEEN ESTABLISHED?
PROMPT18:SET AIR REGISTER TO NORMAL OPERATING POSITION
PROMPT19:PRESS PgDn TO ENGAGE FF CONTROLLER
PROMPT20:MAKE SURE THAT IGNITION IS NOT LOST
PROMPT21:SHUT DOWN IGNITOR, VERIFY THAT FLAME IS MAINTAINED
PROMPT22:SHUT OFF THE BURNER HEADER ATMOSPHERIC VALVE
PROMPT23:HAS A STABLE FLAME BEEN ESTABLISHED?
PROMPT24:RAMP DOWN AF TO ABOUT 6% EXCESS O2
PROMPT25:PRESS PgDn TO ENGAGE O2 CONTROL LOOP
PROMPT26:IS THE BURNER OPERATING PROPERLY?
PROMPT27:PRESS PgDn TO ENGAGE DP CONTROL LOOP
PROMPT28:STARTUP PROCEDURE DONE!
PROMPT30:RAMP DOWN DRUM PRESSURE TO BRING THE BOILER OFF LINE.
PROMPT31:PRESS PgDn TO DISENGAGE DP LOOP CONTROLLER
PROMPT32:RAMP DOWN FF Setpoint UNTIL AF REACHES ITS PURGE RATE
PROMPT33:PRESS PgDn TO DISENGAGE O2 LOOP CONTROLLER
PROMPT35:RAMP DOWN FF TO A STARTUP LEVEL
PROMPT36:PRESS PgDn TO DISENGAGE FF CONTROLLER
PROMPT37:SET DAMPER TO A STARTUP POSITION
PROMPT38: CLOSE THE SAFETY SHUTOFF VALVE TO SHUT DOWN BURNER
PROMPT39:OPEN ALL ATMOSPHERIC VENT VALVES
PROMPT40:COMPLETE A UNIT PURGE
PROMPT41:PRESS PgDn TO DISENGAGE WF & WL CONTROLLERS
PROMPT42:SHUT DOWN BLOWERS AND CLOSE DAMPERS
PROMPT43:SHUTDOWN PROCEDURE DONE!
SMODE0:WAITING INPUT
SMODE1:STARTUP NOW !
SMODE2:SHUTDOWN NOW!

APPENDIX F: Database for Efficiency Calculation

TYPE: AO

TAG............ FUEL
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0
EH-HI EGU...... 5
ET-EGU TAG..... NO
OR-OUT REVERSE. N
CS-COLD START.. 2
DESC.... Fuel Selection
LL-LO OP LIMIT. 0
LH-HI OP LIMIT. 5
LR-RATE LIMIT.. 5
NX-NEXT BLK....
CP-COPY TO.....

TAG............ TEMP
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 100.0
EH-HI EGU...... 1000.0
ET-EGU TAG..... No
OR-OUT REVERSE. N
CS-COLD START.. 300.0
DESC.... Flue Gas Temperature
LL-LO OP LIMIT. 100.0
LH-HI OP LIMIT. 1000.0
LR-RATE LIMIT.. 900.0
NX-NEXT BLK....
CP-COPY TO.....

TAG............ O2JUNK
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 20.00
ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START.. 5.00

DESC.... o2 for test
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 20.00
LR-RATE LIMIT.. 20.00
NX-NEXT BLK....
CP-COPY TO.....

TAG............ EFF
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START..
DESC.... Calculated Combustion Efficiency
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 100.00
LR-RATE LIMIT.. 100.00
NX-NEXT BLK.... EFF-TR
CP-COPY TO.....

TAG............ CO2
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START..
DESC.... calculated CO2
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 100.00
LR-RATE LIMIT.. 100.00
NX-NEXT BLK....
CP-COPY TO.....

TAG............ NG-O2
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 20.00

ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START..
DESC.... O2 FOR CALCULATING NG-CO2 & NG-EFF
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 20.00
LR-RATE LIMIT.. 20.00
NX-NEXT BLK.... NG-CO2
CP-COPY TO.....

TAG............ OIL2-O2
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 20.00
ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START..
DESC.... O2 FOR CALCULATING OIL2-CO2 & OIL2-EFF
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 20.00
LR-RATE LIMIT.. 20.00
NX-NEXT BLK.... OIL2-CO2
CP-COPY TO.....

TAG............ OIL6-O2
DV-DEVICE...... DUMMY
HT-H/W OPTIONS.
IO-ADDR........
SC-SIG COND....
EL-LO EGU...... 0.00
EH-HI EGU...... 20.00
ET-EGU TAG..... PCT
OR-OUT REVERSE. N
CS-COLD START..
DESC.... O2 FOR CALCULATING OIL6-CO2 &OIL6-EFF
LL-LO OP LIMIT. 0.00
LH-HI OP LIMIT. 20.00
LR-RATE LIMIT.. 20.00
NX-NEXT BLK.... OIL6-CO2
CP-COPY TO.....

TYPE: TR

TAG............ EFF-TR
EL-LO EGU...... 45.00
EH-HI EGU...... 95.00

```
ET-EGU TAG..... PCT
AT-AVG/COMPRESS.. 1
NX-NEXT BLK....

TYPE: CA

TAG............ OIL2-CO2
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (B-(C*A))
B - INPUT 2.... 15.611
C - INPUT 3.... 0.7425
D - INPUT 4.... 0.0000
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00
G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK.... OIL2-EFF1
CP-COPY TO.....


TAG............ OIL2-EFF1
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (((((B-(C*H))+((D*H)*H))-(E*A))+((F*A)*A))+((G*A)*H))
B - INPUT 2.... 202.08
C - INPUT 3.... 9.6511
D - INPUT 4.... 0.1732
E - INPUT 5.... 13.217
F - INPUT 6.... 0.4135
G - INPUT 7.... 0.5903
H - INPUT 8.... O2JUNK
NX-NEXT BLK.... OIL2-EFF
CP-COPY TO.....


TAG............ OIL2-EFF
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (A-(B*C))
B - INPUT 2.... 0.0268
C - INPUT 3.... TEMP
D - INPUT 4.... 0.00
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00
```

G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK....
CP-COPY TO.....

TAG............ NG-CO2
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (B-(C*A))
B - INPUT 2.... 11.801
C - INPUT 3.... 0.5615
D - INPUT 4.... 0.0000
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00
G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK.... NG-EFF1
CP-COPY TO.....

TAG............ NG-EFF1
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = ((((((-B)+(C*H))-((D*H)*H))+(E*A))-((F*A)*A))+((G*A)*H))
B - INPUT 2.... 80.655
C - INPUT 3.... 16.757
D - INPUT 4.... 0.4514
E - INPUT 5.... 29.450
F - INPUT 6.... 1.2529
G - INPUT 7.... -1.4427
H - INPUT 8.... O2JUNK
NX-NEXT BLK.... NG-EFF
CP-COPY TO.....

TAG............ NG-EFF
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (A+(B*C))
B - INPUT 2.... -0.0277
C - INPUT 3.... TEMP
D - INPUT 4.... 0.00
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00

G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK....
CP-COPY TO.....

TAG............ OIL6-CO2
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (B-(C*A))
B - INPUT 2.... 16.495
C - INPUT 3.... 0.7848
D - INPUT 4.... 0.00
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00
G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK.... OIL6-EFF1
CP-COPY TO.....

TAG............ OIL6-EFF1
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (((((B+(C*H))+((D*H)*H))+(E*A))+((F*A)*A))+((G*A)*H))
B - INPUT 2.... 23.291
C - INPUT 3.... -18.325
D - INPUT 4.... 1.0014
E - INPUT 5.... 14.700
F - INPUT 6.... -0.6183
G - INPUT 7.... 0.8204
H - INPUT 8.... O2JUNK
NX-NEXT BLK.... OIL6-EFF
CP-COPY TO.....

TAG............ OIL6-EFF
EL-LO EGU...... 0.00
EH-HI EGU...... 100.00
ET-EGU TAG..... PCT
RN-ROUND ENAB.. Y
OUTPUT = (A+(B*C))
B - INPUT 2.... -0.0272
C - INPUT 3.... TEMP
D - INPUT 4.... 0.00
E - INPUT 5.... 0.00
F - INPUT 6.... 0.00

G - INPUT 7.... 0.00
H - INPUT 8.... 0.00
NX-NEXT BLK....
CP-COPY TO.....

TYPE: PG

TAG............ FUEL-MODE
AD-ALM DEST.... TH
AP-ALM PRI..... L
ST-SCAN TIME... 1
IS-INIT SCAN... ON
IA-INIT A/M.... AUTO
DESC.... Choose efficiency calculation models
00 IF FUEL = 1 GOTO 6
01 NUL
02 IF FUEL = 2 GOTO 10
03 NUL
04 IF FUEL = 3 GOTO 14
05 NUL
06 SETOUT NG-O2 O2JUNK
07 SETOUT CO2 NG-CO2
08 SETOUT EFF NG-EFF
09 GOTO 17
10 SETOUT OIL2-O2 O2JUNK
11 SETOUT CO2 OIL2-CO2
12 SETOUT EFF OIL2-EFF
13 GOTO 17
14 SETOUT OIL6-O2 O2JUNK
15 SETOUT CO2 OIL6-CO2
16 SETOUT EFF OIL6-EFF
17 GOTO 0
18 NUL
19 NUL

DISTRIBUTION

Chief of Engineers
ATTN: CEHEC-IM-LH (2)
ATTN: CEHEC-IM-LP (2)
ATTN: CEMP-ET
ATTN: CERD-L

CECPW
ATTN: CECPW-FU-M 22060

Fort Belvoir 22060
ATTN: CECC-R 22060

US Army Engr Districts
ATTN: Library (40)

US Army Engr Divisions
ATTN: Library (13)

US Army Europe
ATTN: ODCS/Engr 09400

USA AMCCOM 61299
ATTN: AMSMC-RI
ATTN: AMSMC-IS

US Army Materiel Command (AMC)
Alexandria, VA 22333-0001
ATTN: AMCEN-F
Installations: (19)

FORSCOM
Forts Gillem & McPherson 30330
ATTN: FCEN
Installations: (23)

US Army HSC
Fort Sam Houston AMC 78234
ATTN: HSLO-F
Fitzsimmons AMC 80045
ATTN: HSHG-DPW
Walter Reed AMC 20307
ATTN: Facilities Engineer

TRADOC
Fort Monroe 23651
ATTN: ATBO-G
Installations: (20)

USARPAC
ATTN: DPW, Ft. Shafter 96858
ATTN: APEN-A

CRREL 03755
ATTN: Library

CEWES 39180
ATTN: Library

Tyndall AFB 32403
ATTN: HQAFCESA Program Ofc

Naval Facilities Engineering Services Center
ATTN: Library, Code L08A 93043

National Guard Bureau 20310
ATTN: Installations Div

NEESA 93043
ATTN: Code 111C

Defense Distribution Region East
ATTN: DDRE-WI 17070

U.S. Gov't Printing Office 20401
ATTN: Rec Sec/Deposit Sec (2)

Defense Tech Info Center 22304
ATTN: DTIC-FAB (2)

144
06/94

☆ U.S. GOVERNMENT PRINTING OFFICE: 1994—3510-S/00038